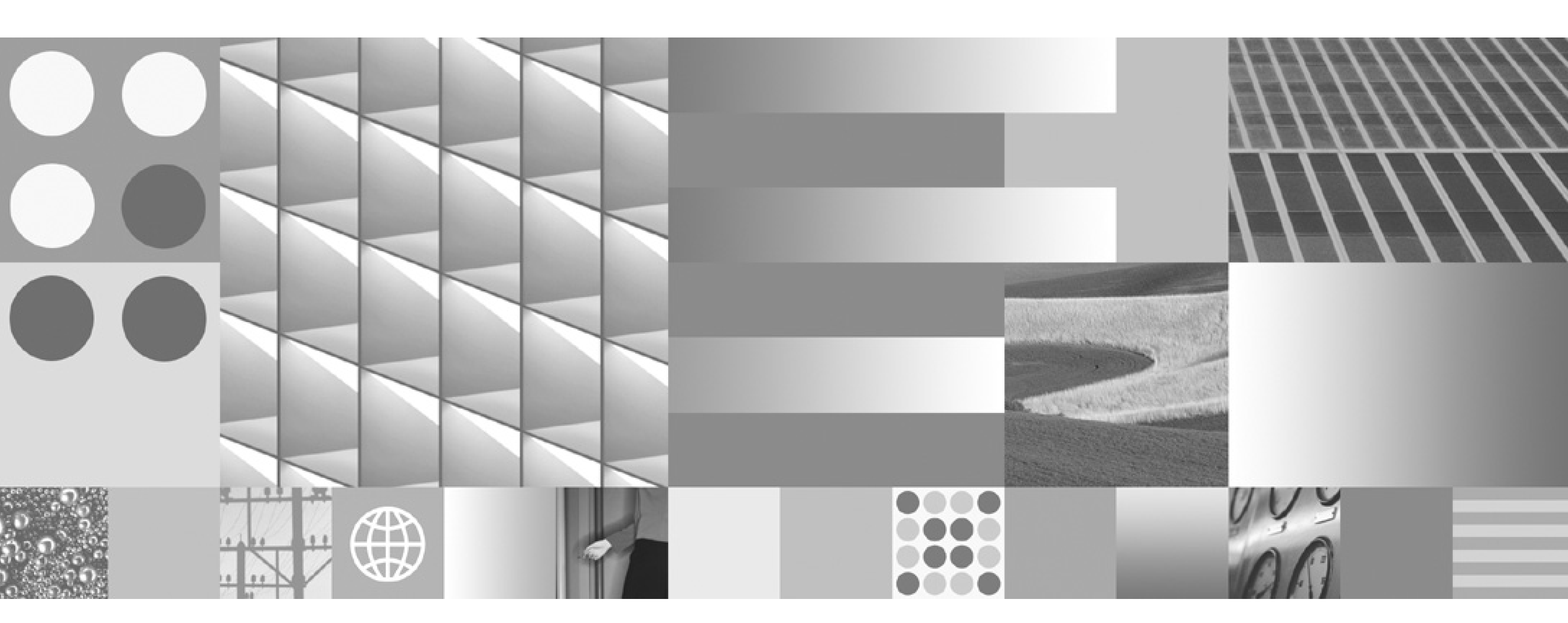
DB2 版本 9.5
Linux 版、UNIX 版和 Windows 版





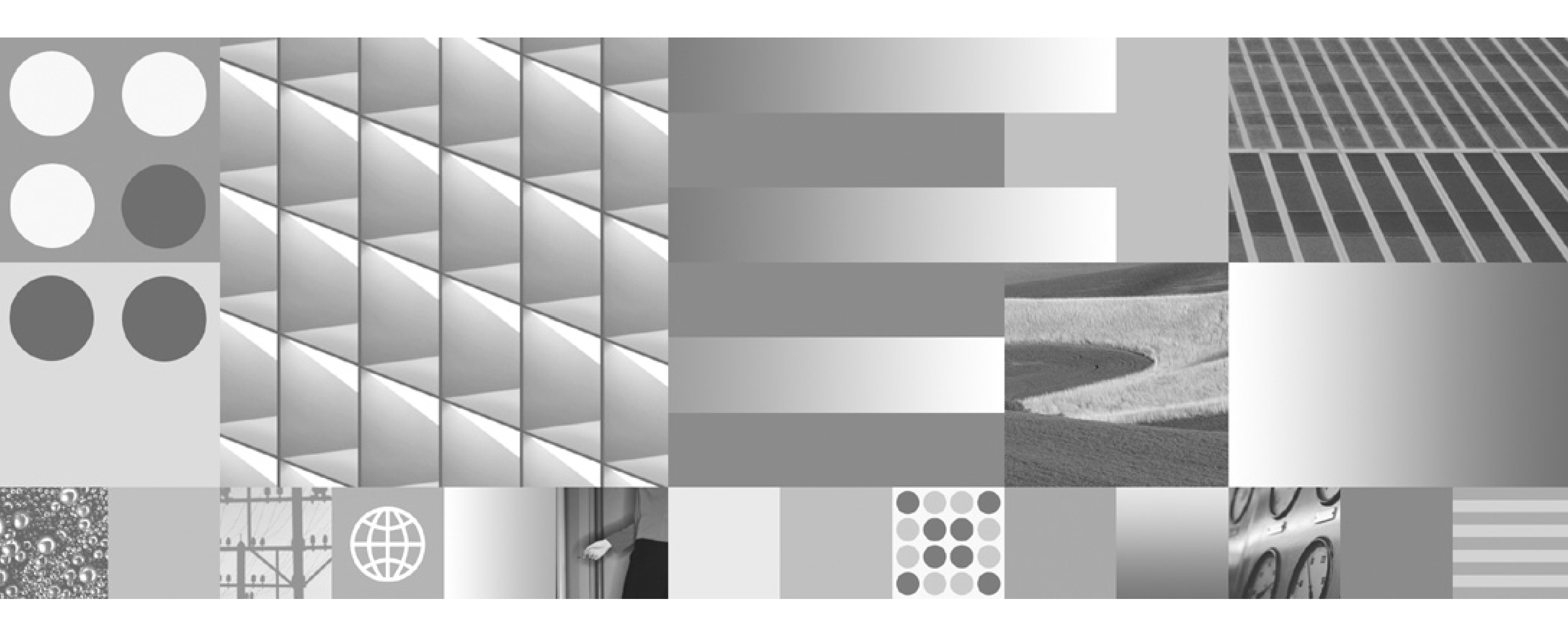
Net Search Extender 管理和用户指南

更新时间：2009 年 4 月

S151-0760-01

DB2 版本 9.5
Linux 版、UNIX 版和 Windows 版





Net Search Extender 管理和用户指南

更新时间：2009 年 4 月

S151-0760-01

注意

使用此信息及其支持的产品前，请先阅读第 225 页的附录 B，『声明』下的常规信息。

修订版声明

此文档包含 IBM 的所有权信息。它在许可协议中提供，且受版权法的保护。本出版物中包含的信息不包括对任何产品的保证，且提供的任何语句都不需要如此解释。

您可在线或通过当地的 IBM 代表处订购 IBM 出版物。

- 要在线订购出版物，请转至 IBM 出版物中心，网址为：www.ibm.com/shop/publications/order
- 要查找当地的 IBM 代表处，请转至 IBM 全球联系人目录，网址为：www.ibm.com/planetwide

要从美国或加拿大的 DB2 市场和销售部订购 DB2 出版物，请致电 1-800-IBM-4YOU（426-4968）。

当您向 IBM 发送信息时，即同意授予 IBM 独一无二的权力以它认为适当且不会对您造成任何影响的方式使用或分发该信息。

目录

第 1 章 Net Search Extender 概述和概念

Net Search Extender 主要概念

对于存储在 DB2、其他数据库和文件系统中的全文本文档，Net Search Extender 为用户和应用程序员提供了一种使用 SQL 查询来搜索这些文档的快速、通用和智能的方法。

要完全了解 Net Search Extender 的功能，需要了解主要术语（在本节中以**粗体**显示）和提供的各种选项。还需要大致了解 DB2® 数据库的概念和术语。

Net Search Extender 实质上搜索的是数据库表的列中保存的**文本文档**。

文本文档必须是可唯一标识的。Net Search Extender 使用表的**主键**来满足此要求。

可以采用各种格式来存储文档。这些格式包括：非结构化纯文本、结构化文本（例如，HTML 或 XML）或者专用文档格式（例如，PDF 或 Microsoft® Office 文档格式）。对于后者，需要必须单独获取其许可证的其他过滤软件。

Net Search Extender 在查询时并不会按顺序搜索所有文本文档（这将花费大量时间），它会创建**文本索引**来搜索文档，这样搜索的效率更高。

文本索引由从文本文档中抽取的重要**词条**组成。

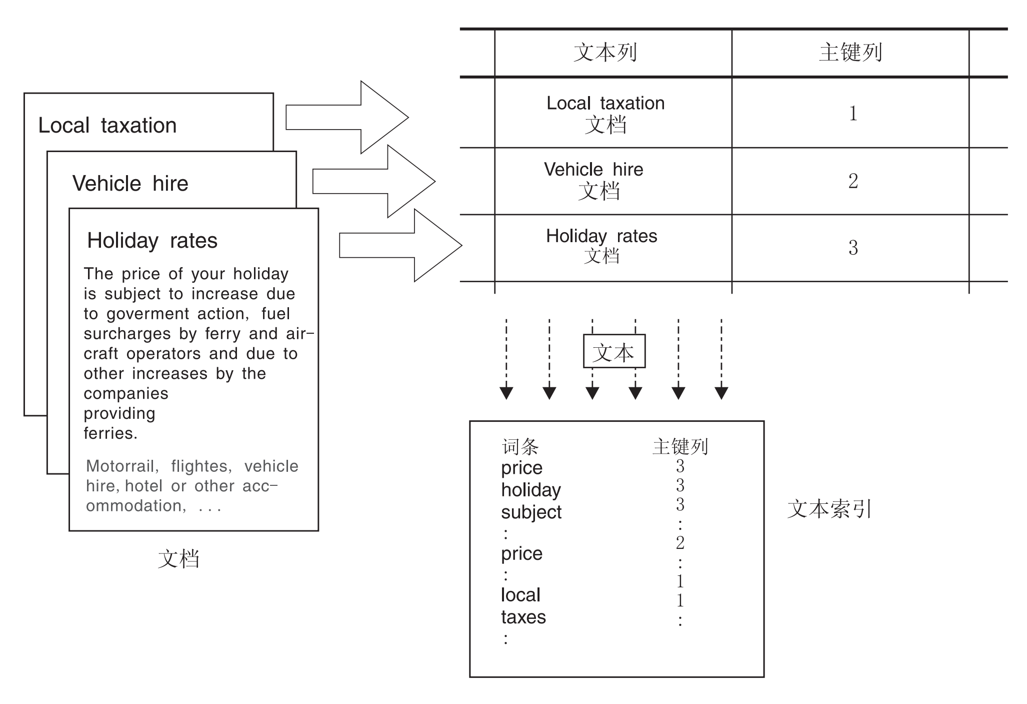


图 1. 创建文本索引

创建文本索引就是定义和声明索引属性（例如，索引的位置）这样一个过程。创建文本索引之后，它并不包含任何数据。**更新索引**是将有关术语和文档的数据添加到文本索引的过程。第一次更新索引时会将文本列中有关所有文本文档的信息添加到索引中。第一次更新被称为**初始更新**。

当使用文本索引进行搜索时，必须考虑表与文本索引之间的同步问题，这是因为必须在文本索引中反映对表进行的任何后续更改（例如，添加、删除和更新文本文档）。

Net Search Extender 中的同步是根据**触发器**来进行的，触发器会将有关新文档、已更改文档和已删除文档的信息自动存储在**日志表**中。每个文本索引都有一个日志表。将日志表中的更改信息应用于相应的文本索引的过程就称为**增量更新**。

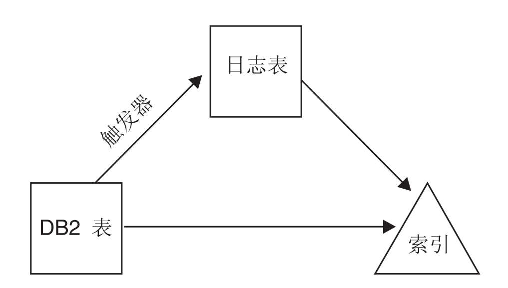


图 2. 增量更新过程

可以使用**手动**或**自动**选项来更新文本索引。自动选项将在指定的日期和时间进行更新。

注意，这些选项**都不会**使更新、删除和插入文本文档的事务范围内的文本索引同步。Net Search Extender 异步建立文本索引提高了性能和并发性。在独立事务内对很小一部分索引的副本应用更新。在用副本代替原始索引的很短时间内，会锁定索引，从而只允许进行读访问。这对搜索操作不可见。有关信息，请参阅第 23 页的『Net Search Extender 实例服务』。

文本索引具有某些属性，例如，索引文件位置和自动更新属性。必要时可以更改某些属性。这称为**改变**索引。改变索引时不会修改任何索引数据。

ORDER BY 短语是否应根据表列对文本索引进行预先排序就是这样一种属性。在这种情况下，初始更新将按指定的顺序对文本文档建立索引，并且按此顺序返回搜索结果。

例如，可以根据书籍价格来指定预先排序的书籍摘要。当查找有关关系数据库系统的最便宜书籍时，可以限制文本搜索只返回前几本书，因为这些书籍是最便宜的。但是，如果没有预先排序的索引，那么必须搜索所有书籍，并将这些搜索结果与最便宜的书籍连接起来，这种操作成本会更高。

Net Search Extender 允许每个文本列具有多个预先排序的索引。例如，一个索引用于根据出版日期来对书籍进行预先排序，而另一个索引用于根据书籍价格来对书籍进行预先排序。

通常，创建文本索引之后的第一次更新是初始更新，后续更新是增量更新。但是，当使用预先排序的索引时，您会想在更新后仍然保持该顺序。这可通过**更新时重新创建索引**选项来解决，每当执行更新时它就会完全重建该索引。

更新文本索引之后，可以使用下列一个选项来进行搜索:

- SQL 标量搜索函数
- 存储过程搜索
- SQL 表值函数

由于各个搜索选项具有不同的操作特征，因此，在下列各节中对它们进行了说明。

SQL 标量搜索函数概述

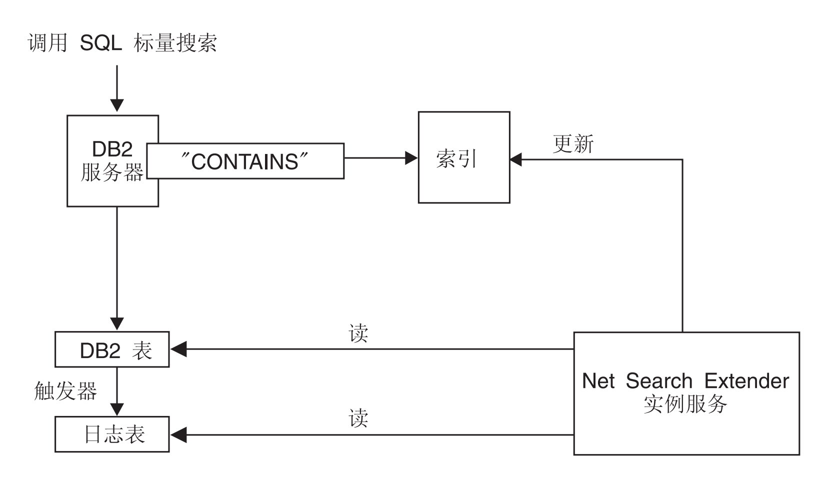


图 3. 使用 SQL 标量搜索函数来进行搜索

Net Search Extender 提供了三个标量文本搜索函数（CONTAINS、NUMBEROFMATCHES 和 SCORE），这些函数无缝地集成在 SQL 中。您可以在 SQL 查询内使用标准 SQL 表达式的相同位置上使用搜索函数。以下是一些典型查询:

```
SELECT * FROM books WHERE CONTAINS (abstract,'"relational databases"') = 1
                    AND PRICE <10

SELECT ISBN, SCORE (abstract, '"relational databases"') as SCORE
                    from BOOKS
                    where NUMBEROFMATCHES (abstract, '"relational databases"')
          >5 AND PRICE <10
                    order by SCORE
```

该示例中的 SQL 标量函数将返回一个指示符，它指示文本文档与给定的文本搜索条件的匹配程度。然后，SQL 查询的 SELECT 部分将确定返回给最终用户的信息。

将 SQL 标量搜索函数用作缺省搜索方法。这些搜索函数应该适用于大多数情况，尤其在将文本搜索表达式与其他条件结合使用时更加适用。

注意: DB2 优化器知道可与 CONTAINS 谓词匹配的文本文档的个数，以及各种备用访问方案在成本上的差异。优化器将选择成本最低的访问方案。

存储过程搜索概述

在向最终用户提供搜索结果时，通常都要调用搜索函数，然后连接用户表，可能还要对结果数据进行排序。执行此操作的成本相当高。然而，在某些情况下，通过对存储在内存中的数据预先进行排序，然后对它们执行连接操作，这样就可以使应用程序不必执行成本很高的磁盘操作。这些情况包括:

- 要提供给用户的数据子集很小
- 已经预先知道数据的子集
- 想要的排序顺序是固定的，并且预先已经知道
- 已进行排名的搜索结果子集足够多

在创建文本索引期间，必须指定要将表或视图外的哪些列返回给最终用户。该数据存储在主存储器的**高速缓存**中。这将使存储过程搜索能够极快地返回搜索结果。需要**激活**高速缓存之后才能使用它，并且还要执行相应的**取消激活**命令。

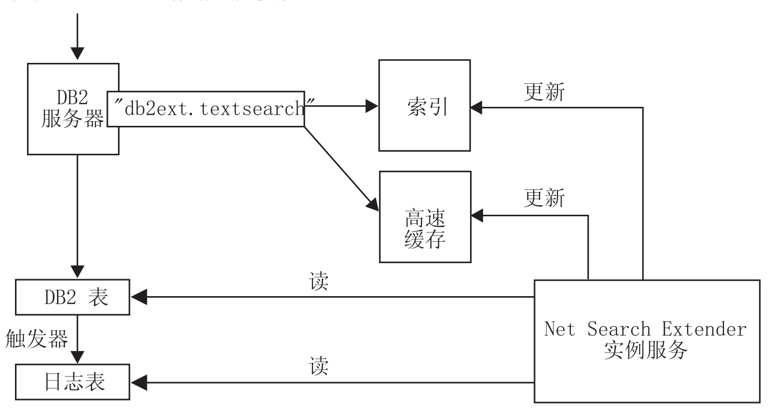


图 4. 使用存储过程搜索

ACTIVATE 命令将数据装入临时高速缓存（它是在激活时从暂存区中创建的）或持久高速缓存（它是在磁盘上维护的）。

如果决定使用存储过程来进行搜索，那么需要仔细计算内存，例如，需要多少内存和应保留多少可用内存用来更新索引。有关缺省值，请参阅第 17 页的『存储过程搜索的内存要求』。

存储过程可以处理对视图创建的文本索引。但是，由于不能对视图创建触发器，因此，不能自动识别任何更改。必须将已更改的信息手动添加至日志表，或者使用 RECREATE 选项。

存储过程搜索主要用于期望执行“仅搜索文本”查询的高性能和高可伸缩性应用程序；“仅搜索文本”查询是指不需要将文本搜索结果与其他复杂 SQL 条件的结果进行连接的查询。

存储过程搜索与 SQL 标量搜索函数在功能方面的主要区别是:

- 不能在任意 SQL 查询中使用存储过程搜索，它只能对预定义的高速缓存表进行查询。
- 存储过程搜索可以利用对视图创建的索引。
- 存储过程搜索可以利用对某一列创建的已预先排序的多个文本索引。

SQL 表值函数概述

SQL 表值函数是 SQL 标量搜索函数与存储过程搜索之间的一种折中方法。借助 SQL 表值函数，还可以使用 db2ext.highlight 函数来获取有关将文档限定为搜索结果的原因的信息。

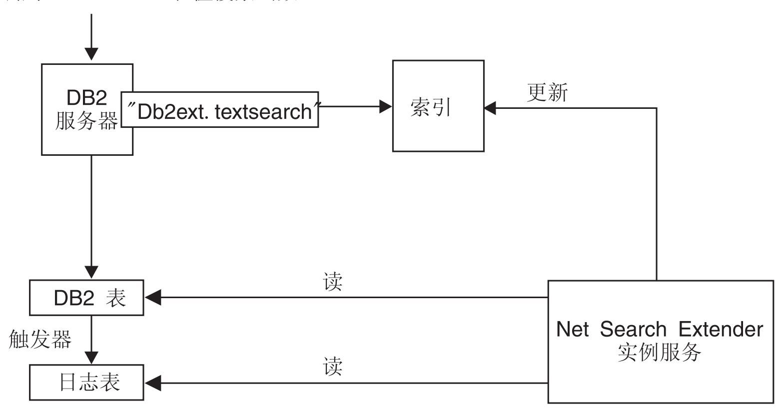


图 5. 使用 SQL 表值函数来进行搜索

SQL 表值函数搜索与存储过程搜索在功能方面的主要区别是:

- 不需要高速缓存（实际上也没有使用任何高速缓存）。
- 在任何 SQL 语句中都可以使用表值函数。
- 不需要大量内存用来预先存储高速缓存表内容。

与 SQL 标量搜索函数在功能方面的主要区别是:

- SQL 表值函数可以利用有关视图的索引。

在某些情况下，通常您会使用 SQL 标量函数，但是如果想利用视图的文本索引，那么可以使用 SQL 表值函数。

其他概念

列变换函数

可以使用您自己的函数来将不受支持的格式或数据类型转换为受支持的格式或数据类型。通过指定“用户定义的函数”（UDF），就可以将原始文本文档作为输入内容。UDF 的输出内容应该是在建立索引期间可以处理的受支持格式。

还可以使用此功能对存储在不直接受支持的外部数据存储器中的文档建立索引。在这种情况下，DB2 列包含文档参考，函数将返回具有相关文档参考的文档内容。

实例服务

“Net Search Extender 实例服务”处理特定于索引的锁定服务和文本索引更新服务（自动和手动）。

存储在外部的数据

在大多数情况下，对其创建文本索引的数据都存储在本机 DB2 表列（例如，`CLOB` 或 `VARCHAR`）中。

但是，也支持存储在外部的文本文档，例如，存储在其他数据库中的文本文档。对于存储在其他数据库中的文档，使用 DB2 昵称表来创建文本索引。

还可以对存储在不受支持的外部数据存储器中的数据使用列变换函数。

管理表和视图

Net Search Extender 中提供了若干种表和视图。它们提供了有关文本索引及其属性的信息。

分区数据库支持

Net Search Extender 的搜索函数采用下列方法来使用分区数据库支持:

- SQL 标量函数利用对分区表创建的索引。
- 存储过程搜索和 SQL 表值函数只利用分区环境中的一个节点上的表。
- 仅在 AIX® 上提供了分区数据库支持。

要启用分区数据库支持，应考虑下列事项:

- 确保按 DB2 文档中所述完成了 DB2 安装。必须使用 root 用户访问权和 setuid 来配置 NFS 安装。
- 在搜索期间，对于每个节点都进行 RESULT LIMIT 求值。这就意味着如果将 RESULT LIMIT 指定为 3 并使用 4 个节点，那么最多可以获得 12 个结果。
- 如果在执行 `db2text start` 期间发生了问题，那么不会返回详细消息来指出受到影响的那些可用节点。如果再次发出 `db2text start`，系统就会尝试对每个节点启动该服务。如果显示了以下消息，那么表示成功执行了 `db2text start` 命令: CTE0185 更新和锁定服务已处于活动状态。

联合数据库中昵称的索引

还可以对联合数据库中的昵称创建文本索引，该索引指向远程数据库中的一些表。在这种情况下，日志表的角色（对于增量索引更新）不同于它对于常规表的索引的角色。与常规表不同，不能对昵称创建 DB2 触发器，所以不能使用触发器将有关文档的更改信息插入到日志表中。因此，对昵称创建索引的增量更新有两种不同方法:

- 日志表是在联合数据库中以本地方式创建的，应用程序将负责使日志表中包含正确的昵称更改信息。对于 DB2 视图，这与增量索引更新相似。此选项是缺省选项。
- 对 DB2 复制进行了设置，这样昵称引用的表更改将被捕获到 DB2 远程数据库的所谓"更改数据表"（CD 表）或非 DB2 关系数据库的"一致更改数据表"（CCD 表）中。于是 DB2 Net Search Extender 可以使用 CD 表或 CCD 表，而不必为昵称的索引创建日志表。在这种情况下，必须在 DB2TEXT CREATE INDEX 命令中指定捕获表特征。

支持本机 XML

由于所有 Net Search Extender 搜索函数都完全支持 SQL XML 数据类型，因此，可以将它们用于存储在数据库中 XML 文档。

通过对搜索结果进行功能强大的 XQuery 处理，就可以扩展按节对 XML 文档进行结构文本搜索这一功能（有关**节**进行搜索的更多信息，请参阅第 138 页的『搜索自变量』）。可以在 DB2 的 XQuery 语言支持中利用 Net Search Extender 文本搜索功能来对 XML 文档进行最佳处理。

通过在 `db2-fn:sqlquery()` XQuery 输入函数中使用全文本搜索，就可以搜索 XML 文档并使用 XQuery 来处理获得的 XML 文档:

```
FOR $dept in db2-fn:sqlquery('select Department from MyTable
    where contains(Department,''sections(/dept/employee/resume) "DB2 XML" '')
      = 1')/dept
RETURN $dept/employee/name
```

在上述示例中，“Department”列的数据类型为“XML”。有关更多信息，请参阅第 81 页的第 8 章，『使用结构化文档』。

提供了样本来说明可以如何查询 XML 数据。请参阅 <sqllib>/samples/extenders/db2ext。调用 xmlsample <database> 以填充数据库并创建和更新索引。在连接至数据库之后，可以通过发出 db2 -tvf xmlsearch 来对数据执行搜索。

DB2 Net Search Extender 的主要功能

Net Search Extender V9.5 具有下列主要功能:

- 建立索引
  - 对大量数据快速建立索引
  - 动态更新索引
  - 可选：在建立索引时将表列存储在主存储器中，以避免在搜索时执行大量的物理读取操作
  - 选择命令行或 DB2 控制中心界面来维护索引
  - 支持结构化文本格式，例如，HTML 和 XML
  - 支持第三方过滤软件“Outside In”
  - 支持昵称表
  - 支持预先排序的文本索引
  - 支持分区数据库（仅适用于 AIX）
  - 支持本机 XML
- 搜索
  - 布尔运算
  - 对同一句子或段中的单词执行相近搜索
  - 对与搜索词条拼写相似的单词进行“模糊”搜索
  - 通配符搜索，对整个单词和单个字符在前面、中间和末尾使用屏蔽
  - 自由文本搜索。对于包含特定文本的文档，搜索自变量是用自然语言表示的
  - 突出显示功能，用来显示将特定文档限定为搜索结果的原因
  - 同义字支持
  - 限制只对文档内的各节进行搜索
  - 数字属性支持
  - 许多并发用户对大量文本文档进行高速搜索
  - 使用 db2-fn:sqlquery() 函数集成到 XQuery 处理中。
- 搜索结果
  - 可以在建立索引时指定如何对搜索结果进行排序
  - 可以在搜索大量数据时和期望大型结果列表时指定搜索结果子集
  - 可以使用高匹配项计数对搜索词条设置限制
  - 与 DB2 优化器组合在一起的内置 SQL 功能将根据期望的搜索结果自动选择最佳方案

介绍 db2text 命令

以下是 Net Search Extender 命令的一个示例:

```
db2text ENABLE DATABASE FOR TEXT
```

对于每个创建和索引维护命令，可以指定数据库、用户和密码。

```
db2text ... connect TO <database> USER <userID> USING <password>
```

注: 如果在 db2text 命令中忽略了连接选项，那么由 DB2DBDFT 环境变量指定数据库。

要显示命令列表，输入以下命令:

```
db2text ?
```

要显示单个命令的语法，输入以下命令:

```
db2text ?command
```

例如，要显示 CREATE INDEX 命令的语法，使用以下命令:

```
db2text ?CREATE INDEX
```

如果已经成功处理了该命令，那么 db2text 将返回 0，如果尚未处理该命令，那么将返回 1。注意，如果存在文档错误，但是索引仍然要更新，那么 db2text 命令将返回 0 和一条警告消息。可以在索引的事件表中找到有关文档错误的信息。

根据操作系统和活动命令 shell，系统将解释诸如 ?、(、)、*、! 和 ″ 之类的特殊字符。因此，如果命令中包含这些字符，应使用引号或转义字符。

以下是使用特殊字符的 UNIX® 命令的一个示例:

```
db2 "SELECT * FROM sample WHERE CONTAINS (DESCRIPTION, '\"enable\"') = 1"
```

第 2 章 安装

在 DB2 客户机/服务器环境中安装 Net Search Extender

Net Search Extender 搜索功能已集成到 SQL 中并在服务器上执行。因此，不需要在客户机上安装 Net Search Extender 以发出文本搜索查询。

如果你计划从远程客户机管理 Net Search Extender，那么必须具有 DB2 服务器的其中一个版本并且 Net Search Extender 已经被安装在客户机端上。

Net Search Extender 安装系统要求

对支持 Net Search Extender 的平台的最低硬件和软件要求与 DB2 V9.5 相同。

有关 DB2 V9.5 安装要求的当前最新列表以及特定操作系统补丁，请参阅 V9.5 信息中心中的『DB2 数据库产品的安装要求』主题，网址为 http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/r0008857.html。

您需要安装 DB2 V9.5 才能运行 Net Search Extender。

Net Search Extender 在以下平台上受支持:

- AIX（64 位）平台
- Linux® x86（32 位）平台
- Linux x86-64（64 位）平台
- Linux on zSeries®（64 位）平台
- 基于 Itanium 的 HP Integrity Series（64 位）平台上的 HP-UX
- Solaris UltraSPARC（64 位）平台
- Windows® on x86（32 位）平台
- Windows on x86-64（64 位）平台

典型 Net Search Extender 安装的最低磁盘空间要求是 50 MB。可能需要额外的硬件要求，这取决于计划建立索引的数据的数量。

分区 DB2 服务器的安装概述（仅 AIX）

确保在每个节点上正确安装和配置 DB2。安装 DB2 之后，需要在每个节点上安装 Net Search Extender。

注: 如果受防护用户标识与实例所有者标识不同，那么它不可使用分区数据库。

在 UNIX 上进行安装

要在 UNIX 上进行安装，执行下列步骤:

1. 安装产品。
2. 更新 DB2 实例。

UNIX 上的第一步: 安装产品组件

要在 UNIX 上进行安装，执行下列步骤:

1. 在目标机器上作为 root 用户登录。
2. 切换到您使用的平台上的正确目录:
   - `cd /<cdrom>`，其中 `<cdrom>` 是 CD-ROM 驱动程序路径。
   - `cd <platform>`
3. 调用 `./nsesetup.sh`，并遵循屏幕上显示的指示信息。

   在接受许可协议之后，就会出现可以使用的安装路径的列表。适合您使用的安装路径取决于想要安装的 Net Search Extender 版本和已安装的 DB2 副本。在选择安装路径之后，就会将产品安装在此路径中。并且会将安装日志文件写入 `/tmp` 目录中，其前缀为 `db2nsei`。

UNIX 上的第二步: 更新 DB2 实例

要更新 DB2 实例，执行下列步骤:

1. 确保您是 root 用户。
2. 使用以下命令来将工作目录切换至 Net Search Extender 的安装路径:

   ```
   cd <path>/instance
   ```

   `<path>` 是安装了 Net Search Extender 的 DB2 副本所在的路径。
3. 使用 `./db2iupdt <db2instance>` 运行 `db2iupdt`，其中 `<db2instance>` 是想要与 Net Search Extender 配合使用的现有 DB2 实例用户标识的名称。

在 Windows 上进行安装

此方法使用一些命令行选项。指定需求参数的命令行选项，并且该选项和其参数之间没有空格。要在 Windows 上进行安装，必须使用具有管理员权限的用户标识登录，然后，执行下列步骤:

1. 使用 `<cdrom>:\windows\install\setup.exe` 将文件从程序包传送至目标机器。

   注意，对于每种 DB2 服务，必须输入用户标识和密码来创建正确的 Net Search Extender 服务。

   在接受许可协议之后，就会出现可以使用的 DB2 副本名称的列表。适合您使用的 DB2 副本名称取决于想要安装的 Net Search Extender 版本。在选择 DB2 副本名称之后，就会将产品安装在此 DB2 副本的安装路径中。

   对于静默安装，可以采用下面两种方式来调用 `setup.exe`:

RECORDMODE
创建静默安装响应文件。

预定义的名为 setup.iss 的响应文件位于安装源目录中。如果想创建新的响应文件，请运行 setup.exe -r。这就会在 Windows 目录中创建新的 setup.iss 文件。将此 setup.iss 文件复制到安装源目录中。但是应确保已备份旧的响应文件。

示例：setup.exe -r -f1"d:\some_directory\setup.iss"

SILENTMODE
静默安装。

确保 setup.iss 文件位于安装源目录中。运行 setup.exe -s。在安装源目录中的 setup.log 文件中，如果 ResponseResult 设置为 0，那么表示安装已成功。

示例：setup.exe -s -f1"d:\some_directory\setup.iss" -f2"d:\another_directory\mysetup.log"

安装参数:

- -r: 记录方式（记录静默安装的响应文件）
- -s: 静默安装（静默地运行安装）
- -x: 卸载方式（移除当前安装）
- -f1: 指定可选响应文件名称（完整路径）
- -f2: 指定可选设置日志文件名称（完整路径）

2. 在传送数据之后重新引导系统。
3. 调用 db2text start 来启动"DB2 Net Search Extender 实例服务"。

每个 DB2 实例都创建一个 Windows 服务。应确保 DB2 实例服务是在用户帐户而不是系统帐户下运行的。

Net Search Extender 不支持 Microsoft Cluster Server。

目录名和文件名

对于所有 Net Search Extender 命令都必须使用 SBCS 字符指定目录名和文件名。路径名（包括文件名）的最大长度为 256 个字节。

安装 Outside In 库

Stellent™ 提供的 Outside In 软件可用于某些平台。有关详细信息，请参阅 http://www.oracle.com

要将 Net Search Extender 与 Stellent™ 提供的 Outside In 软件配合使用，必须为每个平台设置库:

- 在 Windows 上，一定要将这些库所在的目录添加至 PATH 环境变量。
- 在 UNIX 上，将 Outside In 库添加到 DB2 lib 安装目录中。

安装验证

UNIX 上的安装验证

完成下列步骤以验证是否正确安装了 Net Search Extender。

- 执行下列步骤来调用 nsesample 管理脚本以设置文本索引:
  1. 切换至 <instance_owner_home>/sqllib/samples/extenders/db2ext
  2. 调用 ./nsesample <yourdb>。注意，如果尚未存在该数据库，那么此命令会将它创建。
  3. 检查主目录中生成的输出文件 nsesample.log。
- 然后，在同一 DB2 命令窗口中调用一些要执行的样本查询:
  1. 使用 db2 connect to <yourdb> 连接至您的数据库
  2. 使用 db2 -tvf search 执行样本查询
  3. 检查脚本中包含的查询结果。注意，每个查询应当返回一个或多个匹配项。

如果 nsesample.log 文件中不存在任何错误，且所有查询都正在运行，那么说明成功安装了 Net Search Extender。

注: 对于 AIX 操作系统上的分布式数据库，使用以下验证样本:

```
nsesample_partitioned database_name [node_number][table_space_filename]
```

Windows 上的安装验证

完成下列步骤以验证是否正确安装了 Net Search Extender。

- 执行下列步骤来调用 nsesample.bat 管理脚本以设置文本索引:
  1. 调用 db2cmd 以打开 DB2 命令窗口。
  2. 切换至 <sqllib>\samples\extenders\db2ext
  3. 从 DB2 命令窗口中调用 nsesample.bat <yourdb>，其中 <yourdb> 是数据库的名称。注意，如果尚未存在该数据库，那么此命令会将它创建。
  4. 检查当前目录中生成的输出文件 nsesample.log。
- 然后，在 DB2 命令窗口中调用下列要执行的样本查询:
  1. 使用 db2 connect to <yourdb> 连接至您的数据库
  2. 使用 db2 -tvf search 执行样本查询
  3. 检查脚本中包含的查询结果。注意，每个查询应当返回一个或多个匹配项。

如果 nsesample.log 文件中不存在任何错误，且所有查询都正在运行，那么说明成功安装了 Net Search Extender。

卸载 Net Search Extender

要从系统中永久除去 Net Search Extender 并除去所有 Net Search Extender 索引，必须首先禁用包含 Net Search Extender 的每个数据库，然后仅除去 Net Search Extender。

卸载 UNIX 上的 Net Search Extender

在 UNIX 上，完成下列步骤以正确卸载 Net Search Extender。

- 对于想要从其中卸载 Net Search Extender 的每个 DB2 实例:
  1. 切换至 DB2 实例的用户标识
  2. 对于每个数据库都运行以下命令:
     ```
     db2text disable database for text connect to <databasename>
     ```
  3. 停止 DB2 Net Search Extender 实例
  4. 停止 DB2 实例
- 确保您是 root 用户。
- 将工作目录切换至想要从其中除去 Net Search Extender 的 DB2 路径。例如，`cd /opt/IBM/db2/V9.5/install`。
- 发出 `./db2nse_deinstall` 命令。有关命令语法的详细信息，请参阅第 134 页的『db2nse_deinstall 命令』。

卸载 Windows 上的 Net Search Extender

在 Windows 上，完成下列步骤以正确卸载 Net Search Extender。

1. 对每个数据库都运行 `db2text disable database for text connect to <databasename>`。
2. 停止 DB2 实例。
3. 选择**设置 -> 控制面板 -> 添加或删除程序**。从列表中，选择与 Net Search Extender 在安装期间指定的 DB2<COPYNAME> 相关的 Net Search Extender<COPYNAME> 条目。单击**删除**。

第 3 章 迁移至 DB2 Net Search Extender 版本 9.5

迁移至 DB2 Net Search Extender 版本 9.5 要求您先将 DB2 服务器迁移至版本 9.5, 然后使用 db2extmdb 迁移脚本迁移对 Net Search Extender 版本 8 或版本 9.1 中的 Net Search Extender 启用的数据库。

先决条件

- 迁移之前，备份所有文本索引目录和索引子目录。

过程

要迁移至 DB2 Net Search Extender 版本 9.5:

1. 由于 DB2 版本 9.1 或版本 9.5 中不再支持 Data Links Manager，所以在迁移至版本 9.5 之前应执行下列调用:

   ```
   db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT1;
   db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT2;
   db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT4;
   db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT3;
   ```

   注: 如果有些文本索引是使用包含 Data Links 产品功能的较低版本的 Net Search Extender 创建的，那么您仍然能够搜索这些索引，但是将不能更新。如果仍然需要对存储在数据库外部的文本创建索引，那么必须编写一个可以作为变换函数插入到 Net Search Extender 中的 UDF。
2. 使用下列任一任务将安装了 Net Search Extender 的 DB2 服务器从版本 8 或版本 9.1 迁移至版本 9.5:
   - *《迁移指南》*中的『迁移 DB2 服务器（Windows』）
   - *《迁移指南》*中的『将 32 位 DB2 服务器迁移至 64 位系统（Windows）』
   - *《迁移指南》*中的『迁移 DB2 服务器（Linux 和 UNIX）』

   迁移数据库是这些任务的一部分。如果 Linux 或 UNIX 上存在与 DB2 引擎库无关的不受防护外部例程，那么 MIGRATE DATABASE 命令会将这些外部例程重新定义为 FENCED 和 NOT THREADSAFE 并且返回警告消息 SQL1349W。请参阅《迁移指南》中的『迁移 C、C++ 和 COBOL 例程』，以了解有关如何在新的多线程数据库管理器中安全地运行例程的详细信息。模式名为 DB2EXT 且在数据库迁移期间已改变的 Net Search Extender 函数由步骤 5 中的 `db2extmdb` 迁移脚本重新定义为 NOT FENCED 和 THREADSAFE。
3. 安装 DB2 Net Search Extender 版本 9.5。
4. 如果已从 Linux 和 UNIX 操作系统上的 DB2 版本 8 或 DB2 版本 9.1 迁移，请作为 root 用户登录并使用以下语法运行 db2extimigr 脚本:

   ```
   DB2DIR/instance/db2extimigr [-h|-?] InstanceName
   ```

   其中 DB2DIR DB2 版本 9.5 副本的安装目录。
5. 使用以下语法运行 db2extmdb 迁移脚本,以迁移对 Net Search Extender 启用的数据库:

   ```
   db2extmdb <database-name>
   ```

运行此迁移脚本时，不要对具有文本索引的用户表进行任何更改。

所有迁移步骤都记录在名为 db2extm<database-name>.log 的文件中，该文件位于下列其中一个目录中:

- Linux 和 UNIX 操作系统上的 INSTHOME/sqllib/db2ext/
- Windows 操作系统上的 DB2PATH\db2ext\

其中 INSTHOME 是实例主目录，而 DB2PATH 是 DB2 版本 9.5 副本的安装位置。

6. 如果要从 DB2 版本 8 或版本 9.1 的 32 位服务器迁移至 DB2 版本 9.5 的 64 位服务器，那么必须删除文本索引并重新创建文本索引。 在 Net Search Extender 中，不能在 64 位实例中使用在 32 位实例中创建的文本索引。搜索引擎将返回错误 CTE0101 原因码: “17”。
7. 在 Windows 操作系统上迁移之前，如果要使用在 DB2 版本 8 或版本 9.1 副本的安装目录中创建的文本索引，请复原备份的文本索引目录。如果在 DB2 版本 9.5 安装期间选择了迁移操作或者在迁移后卸载了 DB2 版本 8 或版本 9.1 副本，那么需要复原备份的文本索引目录。

   文本索引配置包含迁移前这些文本索引目录的位置。如果不复原文本索引目录，那么使用这些文本索引的查询和索引管理操作将会失败。

第 4 章 规划注意事项

要以最有效的方式使用 Net Search Extender，必须在部署之前进行一些规划。规划可能涉及到几个用户组，包括数据库管理、接口和系统设计人员、系统架构设计师和开发者。

下列各主题对应该考虑的各个方面提供了指导。

- 目录位置和索引存储器
- 表、列和索引名
- 文档格式和受支持的代码页
- Outside In 过滤软件
- 用户角色

有关开发基于 Net Search Extender 的应用程序的更多信息，请参阅下列各主题:

目录位置和索引存储器

需要用于索引的磁盘空间取决于要对其建立索引的数据的数量和类型。指导原则是，对单字节文档建立索引，需要保留大约要对其建立索引的文档大小的 0.7 倍的磁盘空间。对于双字节文档，保留与想要对其建立索引的文档的总大小相同的磁盘空间。总大小可能必须包括存储在通过用户定义的函数检索的活动数据库外部的数据。

工作目录中的临时文件需要的空间量是索引目录中的最终索引文件需要的空间量的 1.0 到 4.0 倍。注意，缺省索引目录为 `../sqllib/db2ext/indexes`，该目录通常位于系统的 `/home` 分区中，并且可能具有大小限制。如果打算创建大型索引，那么务必在一个具有足够磁盘空间的位置显式指定索引和工作目录。

如果您有几个大型索引，应将它们分别存储在不同的磁盘设备上，尤其是在更新或搜索索引期间要同时访问这些索引时更应如此。

要创建、更新和删除 Net Search Extender 索引，可以使用命令行界面，也可以使用 DB2 控制中心。

存储过程搜索的内存要求

对于下列平台，将高速缓存用于存储过程搜索需要大量内存和不同的内存要求:

- AIX
- Windows
- Solaris
- Linux

AIX（64 位）的内存要求

配置系统限制:

- 使用命令 `ulimit -a` 检查系统限制

- 如果具有不属于“不受限制”的值，那么使用下列步骤:
  - 作为 root 用户登录。
  - 备份 `/etc/security/limits` 文件，然后编辑该文件以提高硬限制。
  - 对于所使用的 DB2 实例所有者，将所有值设置为“不受限制”（值为 -1）。

配置共享内存限制:

- 在 AIX 上，不需要配置共享内存限制。

配置交换空间:

- 使用 `lsattr -E -l sys0` 命令获取系统 RAM 大小
- 使用 `lsps -a` 命令获取交换空间大小。
- 将交换空间大小设置为至少是系统的 RAM 数量的 1.5 至 2 倍，或者使用在 `CREATE INDEX` 命令中提供的 `MAXIMUM CACHE SIZE` 参数。使用 SMIT 实用程序来选择较大的数目。

Windows（32 位和 64 位）的内存要求

调整页面调度文件的大小:

- 将 Windows 虚拟内存页面调度文件大小设置为至少是系统的 RAM 数量的 1.5 至 2 倍，或者使用在 `CREATE INDEX` 命令中提供的 `MAXIMUM CACHE SIZE` 参数。选择较大的那一个数目。有关更改页面调度文件大小的信息，请参阅 Windows 文档。

  在 32 位的 Windows 上，建议不要超过最大高速缓存大小，大约为 1000 MB（1 GB = 1073741824 个字节）。

Solaris（64 位）的内存要求

配置系统限制:

- 使用 `ulimit -a` 命令来检查系统限制
- 然后，执行下列步骤:
  - 作为 root 用户登录。
  - 备份 `/etc/system` 文件，然后编辑该文件以提高硬限制。
  - 添加或检查是否将下列行设置为至少是所显示的最小值:

    ```
    rlim_fd_cur -> Default 64, recommended >= 1024
    ```

    ```
    rlim_fd_cur_max -> Default 1024, recommended >= 4096
    ```

配置共享内存限制:

- 使用 `sysdef -i` 命令来检查当前设置
- 编辑文件 /etc/system，以使用 `set shmsys:shminfo_shmmax=0xffffffff` 来设置共享内存大小限制

  可能还必须增大下列参数值:

  `set shmsys:shminfo_shmmni=512`

  `set shmsys:shminfo_shmseg=128`。然后，重新引导系统。

配置交换空间:

- 使用 `/usr/sbin/prtconf` 命令获取系统 RAM 大小
- 使用 `swap -l` 命令获取交换空间大小。
- 将交换空间大小设置为至少是系统的 RAM 数量的 1.5 至 2 倍，或者使用在 `CREATE INDEX` 命令中提供的 `MAXIMUM CACHE SIZE` 参数。选择较大的那一个数目。

  有关如何添加交换空间的信息，请参阅 Solaris 系统文档。

  建议不要超过最大高速缓存大小，大约为 2000 MB（2 GB = 2147483647 个字节）。

Linux（32 位和 64 位）的内存要求

有关 Linux 上的建议内核参数的信息，请参阅 DB2 文档。

新的 Linux 内核和分发的验证状态是频繁更新的。要获取受支持的 Linux 软件级别的最新信息，请参阅 http://www.ibm.com/software/data/db2/linux/validate。

要查看当前共享资源限制，使用 `ipcs -l`。要检查系统限制，使用 `ulimit -a` 命令。

表、列和索引名的注意事项

通常，所有表名、列名和索引名都不分区大小写。Net Search Extender 还允许您采用混合大小写来指定这些名称。在 Windows 上，如果您想采用混合大小写来指定表名、列名和索引名，那么必须用反斜杠（\）和双引号（″）组成的字符序列将这些名称括起来。例如， \″DocTxt\″。

文档格式和受支持的代码页

Net Search Extender 需要知道您打算搜索的文本文档的格式（或类型）。此信息对于为文本文档建立索引是必需的。

Net Search Extender 支持下列文档格式:

TEXT 纯文本（例如，平面 ASCII），通常是没有任何标记的文本

HTML 超文本标记语言

XML 扩展标记语言

文档格式 XML 是列数据类型 XML 的缺省值，也是该数据类型唯一支持的文档格式。

GPP 通用格式（带有用户定义的标记的平面文本）

Outside In（INSO）

如果要使用过滤软件从 PDF 和其他常见文本格式化工具（例如， Microsoft Word）中抽取文本内容，那么使用这种格式。

对于文档格式 HTML、XML、GPP 和 Outside In 过滤器格式，可以限制只搜索文档的特定部分。

在由于文档格式不受支持而不能使用 Outside In 过滤器的情况下，您可以编写一个自己会执行过滤的“用户定义的函数”（UDF）。必须在创建索引时指定此 UDF 并将数据从不受支持的格式转换为受支持的格式。

如果文档采用其中一种受支持的“编码字符集标识”（CCSID）存储的，那么可以对文档建立索引。有关这些代码页的列表，请参阅 DB2 文档。

要检查数据库代码页，使用以下 DB2 命令:

```
db2 GET DB CFG for <dbname>
```

并采用为数据库代码页写入的值。

为了保持一致，DB2 通常将文档的代码页转换为数据库的代码页。但是，当将 DB2 数据库中的数据存储在具有二进制数据类型（例如，BLOB 或 FOR BIT DATA）的列中时，DB2 不会转换数据，且文档将保持其原始 CCSID。

注意，在创建文本索引或进行搜索时，不兼容的代码页可能会导致产生问题。

Outside In 过滤软件

Net Search Extender 支持第三方文档过滤软件。Outside In 变换技术由 Stellent™ 提供，您可以使用该软件从 PDF 文件或从以常见文本格式工具的专有格式编写的文档中抽取文本内容，而无需使用本机应用程序。示例格式包括 Microsoft Word 和 Lotus® Word Pro®。

在执行 UPDATE INDEX 期间，Net Search Extender 将 Outside In 库作为插件装入。这些库不是 Net Search Extender 的一部分，需要单独安装。应确保 Net Search Extender 可以找到 Outside In 库。

Outside In 软件不仅会生成文本内容，而且会生成结构信息（例如，字段）。Net Search Extender 还可以定制要将 Outside In 生成的文档信息的哪一部分存储在索引中。为此，需要应用特定类型的文档模型（Outside In 文档模型）。

要查看过滤格式和受支持平台的列表，请访问 Stellent Web 站点（http://www.stellent.com）。

用户角色

DB2 实例所有者

DB2 实例所有者用户可以启动和停止 DB2 Net Search Extender 的实例服务并控制锁定服务。此外，为 DB2 实例用户授予了对每个已启用的数据库的 DBADM 权限。因此而对由 Net Search Extender 产生的所有数据库更改实现了一个中央控制点。

必需的 DB2 权限

对 DBADM 授予了 ENABLE DATABASE 权限。

必需的文件系统权限

对所有文本索引目录的读写访问权以及对模型文件的读访问权。

实例所有者的命令

DB2TEXT START、DB2TEXT STOP 和 DB2TEXT CONTROL。

这些命令只允许在服务器上运行。在分布式 DB2 环境中，可以在已配置的任何节点上运行这些命令。每个命令都会检查运行该命令的用户是不是 DB2 实例所有者。作为实例所有者，如果您决定使用受防护的用户标识来运行存储过程和

UDF，那么受防护的用户必须对索引目录中的所有文件具有读写访问权（但对整个目录路径具有读访问权）。注意，受防护的用户标识和实例用户标识必须是相同主组的成员，这样才能为实例用户标识授予对受防护的用户标识所创建的文件的正确访问权，反之亦然。指定正确的组成员资格和文件许可权。

数据库管理员

数据库管理员可以启用和禁用数据库与 Net Search Extender 配合使用。

必需的 DB2 权限
DBADM（对于 ENABLE DATABASE 的 SYSADM 权限）。

数据库管理员的命令
`DB2TEXT ENABLE DATABASE` 和 `DB2TEXT DISABLE DATABASE`。

文本表所有者

文本表所有者可以创建、删除和更改索引。注意，他们必须能够控制（即，具有读写访问权）索引的位置和对全文本索引的更新。

必需的 DB2 权限和特权
文本表的所有者。

文本表所有者的命令:
`DB2TEXT CREATE INDEX`、`DB2TEXT DROP INDEX`、`DB2TEXT ALTER INDEX`、`DB2TEXT ACTIVATE CACHE`、`DB2TEXT DEACTIVATE CACHE`、`DB2TEXT UPDATE INDEX`、`DB2TEXT CLEAR EVENTS` 和 `DB2EXTTH`。

注意，使用 DB2 实例所有者的用户标识部分运行命令实现。因此，在创建或改变文本索引之前，为实例所有者授予必需的文件系统访问权。

第 5 章 管理 Net Search Extender

Net Search Extender 实例服务

DB2 Net Search Extender 实例服务由以下服务组成:

- 锁定服务
- 更新服务

以下主题说明了如何启动和停止 DB2 NetSearch Extenter 实例服务，并更进一步详细讨论了锁定服务和更新服务。

- 启动和停止 NSE 实例服务
- 锁定服务
- 更新服务
- NSE 信息目录

使用 DB2 控制中心启动和停止 Net Search Extender 实例服务

在可以维护文本索引和搜索文档之前，必须启动“Net Search Extender 实例服务”。

要启动“实例服务”，登录到 DB2 实例所有者用户标识（仅限于 UNIX 系统），并输入以下命令:

```
db2text start
```

要停止“实例服务”，输入以下命令:

```
db2text stop
```

注意，每个 DB2 实例都必须有一个“Net Search Extender 实例服务”。锁定服务维护对该实例的所有已启用数据库的锁定。

NSE 锁定服务

当启动 Net Search Extender 时，将自动启动锁定服务。在 Net Search Extender 中，要使对文本索引的并发访问同步就需要锁定服务。

锁定服务可以确保没有两个进程同时尝试更改一个文本索引，或者当一个进程正在更改一个文本索引数据时没有其他进程读取该文本索引数据。因此，大多数进程在启动之前将请求锁定文本索引，而在完成处理之后再释放该锁定。

注意，不能将 Net Search Extender 文本索引的锁定服务与用来控制对 DB2 表的访问的 DB2 锁混淆。

使用锁定服务

在 Net Search Extender 中，有几种不同类型的锁定来控制对索引的并发访问。根据是在搜索请求中只读取文本索引，还是需要计算对文本索引的更改，然后将其写入文件中，而使用不同的锁，正如索引更新一样。

在 db2text start 运行期间，将自动启动锁定服务。对文本索引具有下列类型的锁:

S 锁 用于共享只读访问。例如，搜索请求。

U 锁 用于当使用并发读取访问来计算对索引的更改（更新）时进行读写访问。

X 锁 用于较短时间内进行互斥读/写访问，在这段时间内将把更改真正写入索引中。

IX 锁 由预计的互斥读/写访问用于防止在更新进程等待 X 锁时出现任何新的 S 锁。

每个 DB2 实例都有一个 Net Search Extender 锁定服务。锁定服务为多个数据库维护锁。

锁定服务配置文件为 db2extlm.cfg。对于 UNIX 系统，该文件存储在 <instance_owner_home>/sqllib/db2ext 目录中；而对于 Windows，该文件存储在 <sqllib>\<DB2INSTANCE>\db2ext 目录中。

仅当“Net Search Extender 实例服务”是在 db2text start 期间启动时，对配置文件所作的更改才会生效。用户可以设置下列值:

- 数据库的最大数目
- 每个数据库的最大索引数
- 每个索引允许的最大锁数（并发用户）
- 获得锁的等待时间和尝试次数

配置文件的缺省值为如下所示:

```
<default
                  maxDbs         = "  8"
                  maxIdxPerDb    = " 50"
                  maxLocksPerIdx = "100"

                  sWait = "  50"
                  uWait = " 500"
                  xWait = " 500"

                  sAttempt = "50"
                  uAttempt = "10"
                  xAttempt = "60"

                  latchTimeout = "80"

/>
```

语法为 <default attribute=value.../>，属性的含义如下:

maxDbs
: 锁定服务可以处理的数据库数目（大于 1 的整数）。

maxIdxPerDb
: 可以锁定的每个数据库的索引数（大于 1 的整数）。对于所有数据库，此值都是相同的。

maxLocksPerIdx
: 一个索引中可以同时存在的锁数（大于 1 的整数）。对于所有索引，此值都是相同的。

  如果将 maxDbs、maxIdxPerDb 或 maxLocksPerIdx 的值增大为超过以上提到的配置文件中的缺省值，那么应确保具有足够的内存。

sWait/sAttempt

请求 S 锁时，sAttempt 是指不能立即授予该锁时尝试的次数。sWait 是这些尝试之间的等待时间（大于 1 的整数）。这些参数也适用于 IX 锁。

uWait/uAttempt

当请求 U 锁时，uAttempt 是指不能立即授予该锁时尝试的次数。uWait 是这些尝试之间的等待时间（大小 1 的整数）。

xWait/xAttempt

当请求 X 锁时，xAttempt 是指不能立即授予该锁时的尝试次数。xWait 是这些尝试之间的等待时间（大于 1 的整数）。

latchTimeout

这是时间间隔锁定服务的附加等待时间。要确定对某个锁的总等待时间，使用以下计算公式:

```
waiting time = # attempts * (# waits + (2 * # latchTimeout))
```

等待时间是以毫秒为单位计算的。注意，对于每次尝试，latchTimeout 值在添加到总的等待时间时都要加倍。

查看锁快照

可以使用下面的其中一个命令来查看锁快照:

- 对于单个文本索引:

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase INDEX myindex
```

- 对于数据库的所有锁定文本索引:

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase
```

注意，只有实际锁定的索引在列表中。

当第一次锁定文本索引时，就为数据库和锁定服务中的文本索引保留了内存。如果锁定了更多文本索引，那么在锁定服务中还要为这些索引分配内存。仅当删除文本索引或禁用数据库或者重新启动 Net Search Extender 服务时才会释放此内存。这就意味着文本索引或数据库在锁定服务中会消耗内存（即使当前没有设置任何锁）。

命令 `db2text CONTROL CLEAR ALL LOCKS` 将强制释放对数据库或索引的所有锁。有关如何使用此命令的详细信息，请参阅第 101 页的『CONTROL 命令』。注意，此命令不会释放为数据库或索引分配的任何内存。要释放内存，必须删除该索引或禁用数据库，或者重新启动 Net Search Extender 服务。不要在处于活动状态的索引更新过程中释放锁定。

更新服务

表更改与索引更新不同步。可以手动启动索引更新过程，也可以安排按给定的时间间隔自动开始索引更新过程。更新服务提供此功能，并在执行 `db2text start` 期间启动。

如果使用以下命令需要更新索引，那么在创建索引期间可以指定更新服务的检查频率:

```
db2text create index DB2EXT.TITLE for text on DB2EXT.TEXTTAB (TITLE)
        UPDATE FREQUENCY D(1,3) H(0,12) M(0) update minimum 5
```

在此示例中，将在每周星期一和星期三晚上 12 点和上午 12 点唤醒“更新服务”，并检查是否要对索引 `db2ext.title` 完成某些工作。注意，在此示例中，在将启动自动索引更新以使文本索引与数据库同步之前至少需要对 `DB2EXT.TITLE` 进行五个更改。

在分区数据库环境中，更新服务只在一个节点上启动。

注意

如果将索引更新过程的时间间隔设置得非常短，那么会对系统性能产生负面影响。必须考虑您想在每次更新过程中处理的更改数和将花费的时间，还要考虑您想在自动更新索引期间处理的索引数。确保每个索引更新之间的时间间隔足够长，以便在开始进行预定的下一次更新之前就完成了更新，并且没有安排同时开始对若干个索引进行更新。

使用 DB2 控制中心

使用 DB2 控制中心来管理 Net Search Extender 管理函数、DB2 实例、数据库和数据库对象（例如，表、视图和用户组）。

可以对不同的 DB2 控制中心对象调用命令，例如：

- 实例对象
- 数据库对象
- 索引对象

DB2 控制中心的主要元素包括菜单栏、工具栏、对象树和内容窗格。

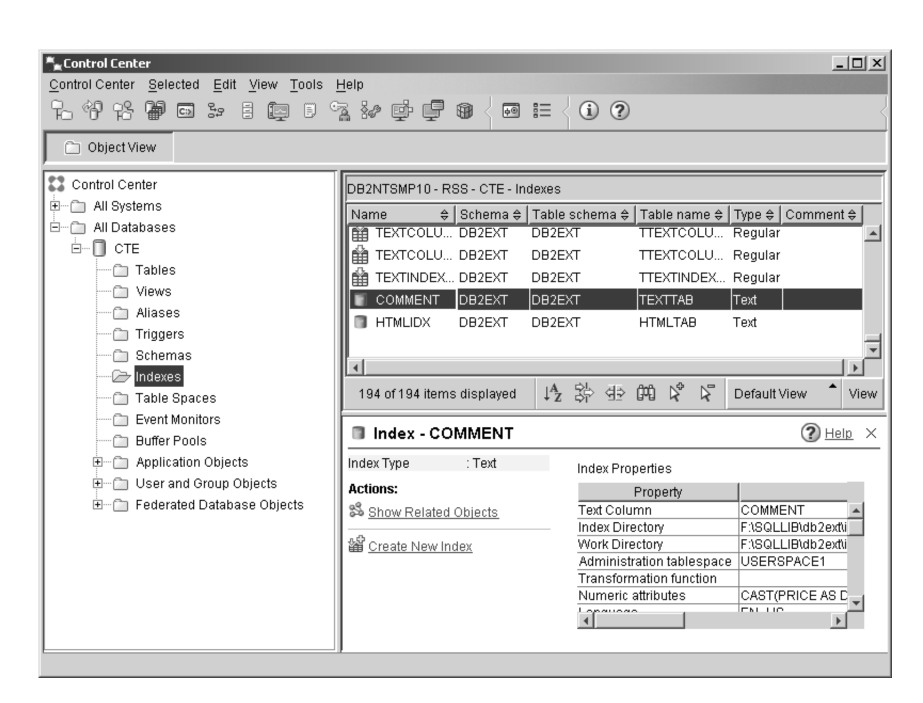


图 6. DB2 控制中心

也可以使用命令行来完成上述任务。有关更多信息，请参阅下列各节：

- 第 43 页的第 6 章，『开发：创建和维护文本索引』
- 第 23 页的『Net Search Extender 实例服务』

注: 在本章中只能找到 Net Search Extender 建立索引函数和管理函数。有关使用 DB2 控制中心来完成其他任务的信息，请参阅 DB2 文档。

使用 DB2 控制中心启动和停止 Net Search Extender 实例服务

从对象树中，单击系统以显示可用的实例。突出显示该实例并进行右键单击以显示实例对象弹出菜单。突出显示 **Net Search Extender**，并从弹出菜单中选择下面的其中一个命令:

启动 Net Search Extender 实例服务
: 如果尚未启动实例服务，那么会启动

停止 Net Search Extender 实例服务

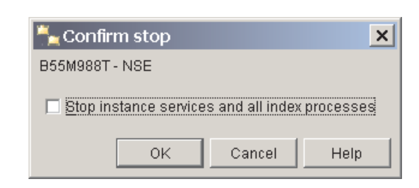


图 7. 停止 Net Search Extender 服务对话框

将显示一个对话框。使用复选框来停止实例服务和索引进程。在命令行语法中，这被称为 `FORCE` 选项。单击**确定**按钮。

实例状态
: 将出现一个对话框显示实例的状态。

使用 DB2 控制中心启用和禁用数据库

从 DB2 控制中心进行数据库管理

在对象树中，单击实例对象以显示可用的数据库。突出显示该数据库并进行右键单击以显示弹出菜单。突出显示 **Net Search Extender**，并从展开的菜单中选择下面的其中一个命令:

启用数据库的文本搜索
: 如果未启用数据库的文本搜索，会显示一个对话框。单击**确定**按钮以启用数据库。如果已启用数据库，那么会显示一个消息框。

禁用数据库的文本搜索
: 如果未禁用数据库的文本搜索，会显示一个对话框。单击**确定**按钮以禁用数据库。如果已禁用数据库，那么会显示一个消息框。

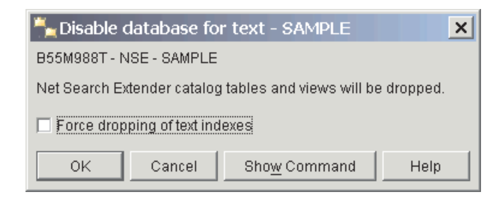


图 8. “禁用数据库的文本搜索”对话框

如果想要禁用数据库并删除所有文本索引，那么单击该复选框。

注意，在所有对话框中，**显示命令**按钮都将显示命令的命令行版本。

使用 DB2 控制中心管理文本索引

在对象树中数据库对象下您可以看到索引对象。单击索引对象可以在内容窗格中查看相应的索引。可以通过“类型”列中的“文本”类型识别文本索引对象。

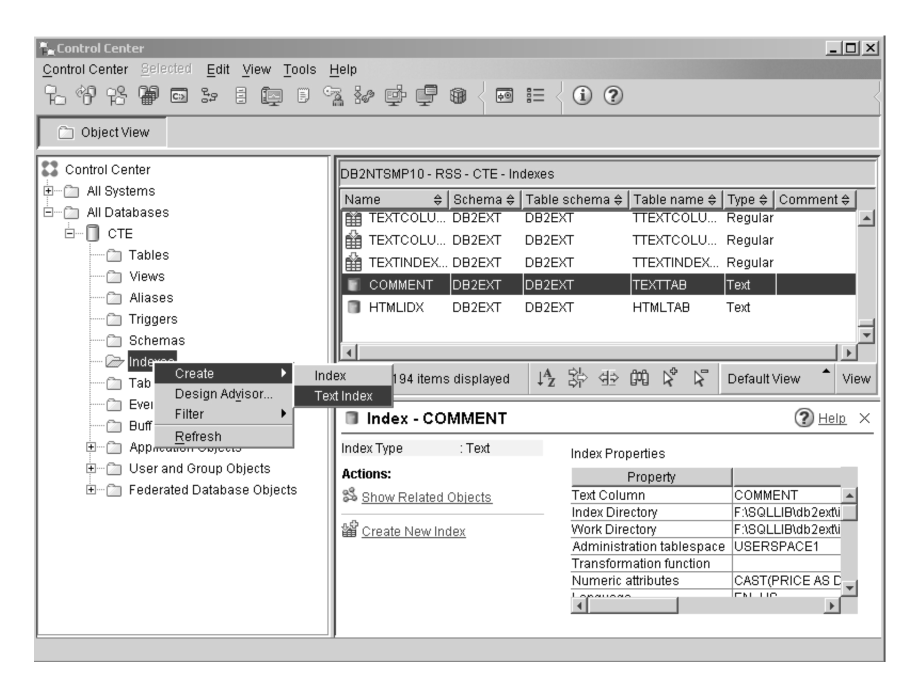


图 9. DB2 控制中心

右键单击索引对象，并从弹出菜单中选择下面的其中一个命令:

创建 将显示一个对话框。通过在对话框中选择**文本索引**，启动用于创建文本索引的向导。

过滤 将显示一个对话框，可在该对话框中选择将哪些索引对象显示在控制窗格视图中。

刷新 将刷新对象树和控制窗格中的信息。

注:

如果不想使用右键单击选项来访问实例、数据库和文本索引对象命令，可单击**所选**菜单命令，然后突出显示 **Net Search Extender** 以访问相关命令。

在创建文本索引之前，应确保已经考虑了第 17 页的第 4 章，『规划注意事项』中的先决条件。

其他建立索引的先决条件包括:

- 启动 Net Search Extender 实例服务
- 启用数据库

使用 DB2 控制中心创建文本索引

选择**创建**命令，并从扩展菜单中突出显示**文本索引**。将弹出“创建文本索引”向导。使用该向导中的数个面板为文本索引指定配置选项。

要在各面板之间移动，应输入所有必填信息，并单击**下一步**按钮，直到**完成**按钮可用为止。单击**完成**按钮即可创建文本索引。

名称面板

用途

此面板允许您指定文本索引的模式和名称。还可以指定文本索引文件的工作目录和索引目录。为管理表空间上的索引创建管理表。

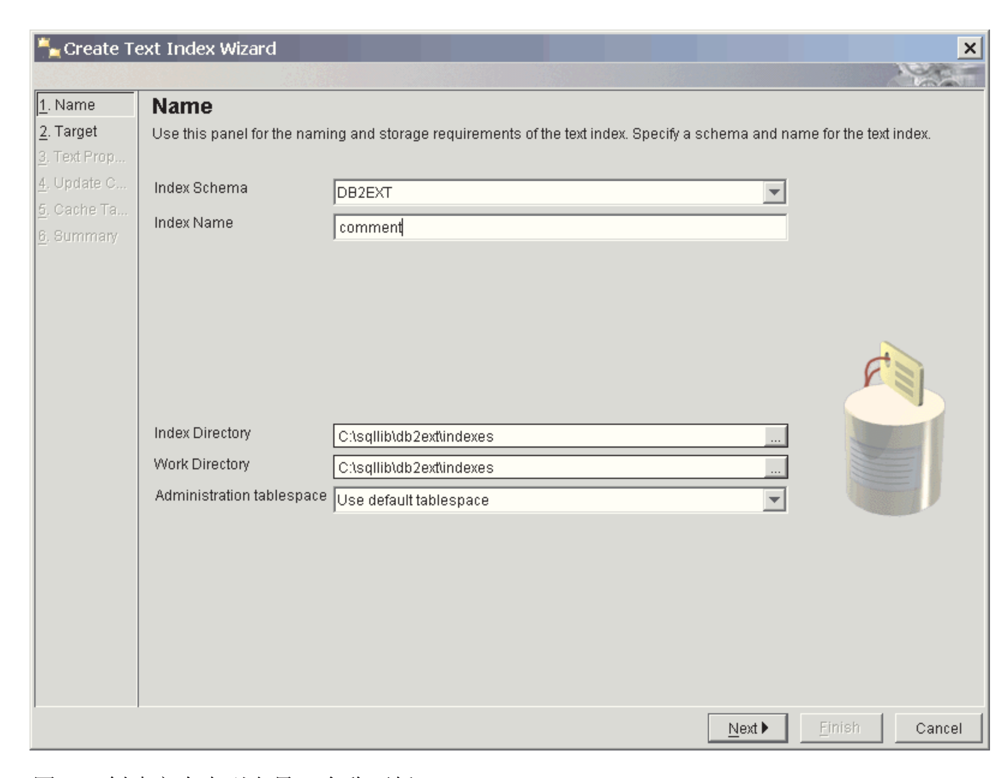


图 10. 创建文本索引向导：名称面板

以下是对该面板中各字段的描述:

表 1. 名称面板文本字段

字段名称	必填/可选	缺省值	描述
索引模式	必填	用户标识	选择文本索引的模式名。这是特定于索引的管理表的 DB2 模式名。
索引名称	必填	无	为文本索引输入有效的 DB2 索引名。借助索引模式，索引名将唯一地标识数据库中的全文本索引。
索引目录	可选	请参阅“路径名”	指定将用来存储文本索引的目录路径。对于 DB2 实例所有者用户标识，目录必须存在，并且具有读、写和执行许可权。

表 1. 名称面板文本字段 (续)

字段名称	必填/可选	缺省值	描述
工作目录	可选	请参阅“路径名”	指定在执行搜索和管理操作期间将用来存储临时文件的工作目录。对于 DB2 实例所有者用户标识，目录必须存在，并且具有读、写和执行许可权。
管理表空间	可选	使用缺省表空间	选择文本索引管理表的表空间名。必须在用户表的表空间所在的同一个节点组上定义管理表空间。

目标面板

用途

此面板允许您指定表或昵称表的模式和名称以及包含想要建立索引的数据的文本列的名称。可以使用变换函数来修改文本列的内容。除了文本列之外，如果想要将表列表达式的内容添加至文本索引，那么还可以指定数字属性。

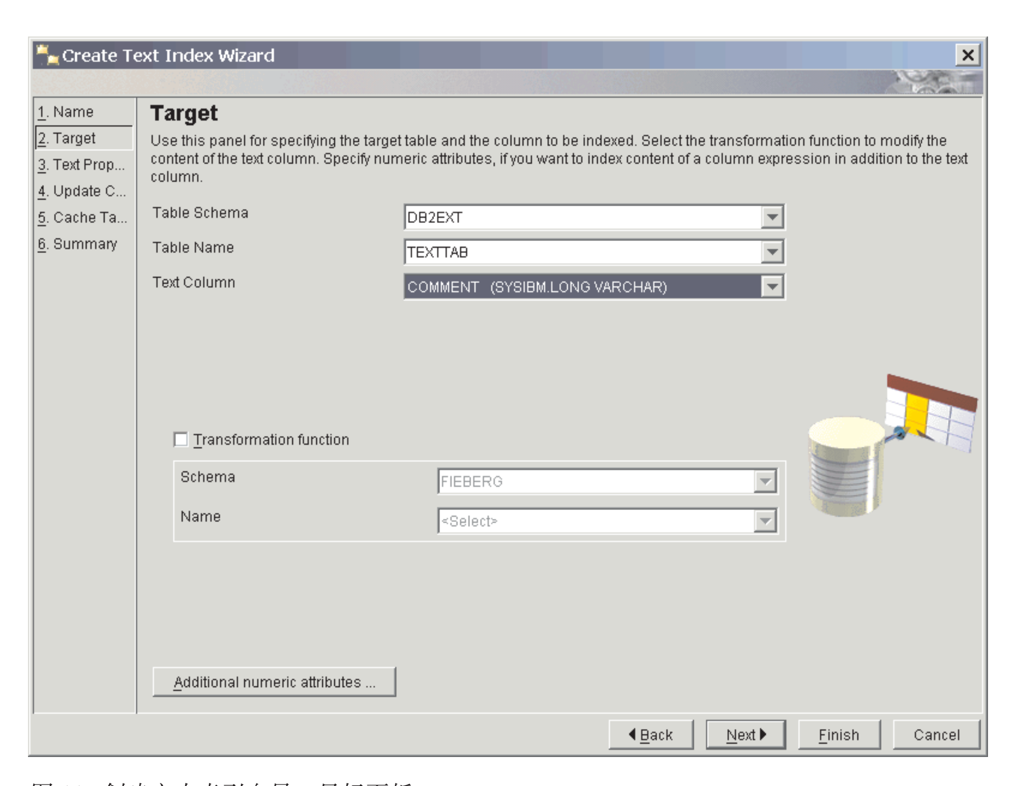


图 11. 创建文本索引向导: 目标面板

以下是对该面板中各字段的描述:

表 2. 目标面板文本字段

字段名称	必填/可选	缺省值	描述
表模式（1）	必填	用户标识	选择正在对其创建文本索引的表或昵称表的模式。
表名（2）	必填	无	选择正在对其创建索引的表或昵称表的名称。表必须具有主键。

表 2. 目标面板文本字段 (续)

字段名称	必填/可选	缺省值	描述
文本列（3）	必填	无	选择用于创建文本索引的列的名称。列必须属于下列类型：CHAR（用于位数据）、VARCHAR（用于位数据）、LONG VARCHAR（用于位数据）、CLOB、DBCLOB、BLOB、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC 和 XML。如果不是这种情况，那么指定的变换函数必须确保传递此处所提到的其中一种有效数据类型。 **注：** 不推荐使用 LONG VARCHAR 和 LONG VARGRAPHIC 数据类型，将来的发行版会将其除去。
变换函数	可选	禁用	选择此项以使用变换函数。
变换函数：模式	如果选择了函数，那么此字段是必填字段	用户标识	选择用来访问文本文档的 UDF 的模式。
变换函数：名称	同上	无	选择用来访问文本文档的 UDF 的名称。

注意，只能按顺序指定表模式（1）、表名（2）和文本列（3）。

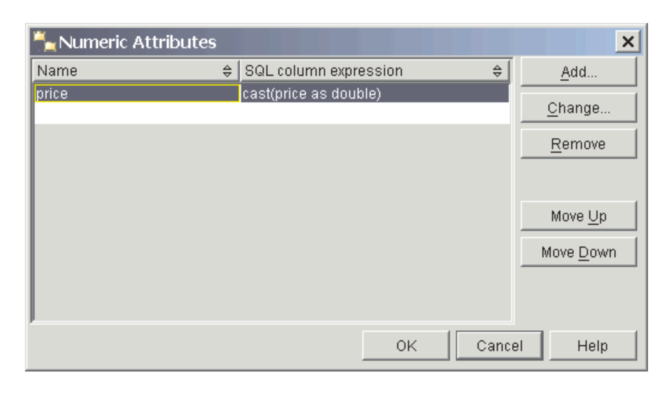


图 12. 数字属性对话框

要查看或添加属性，单击**数字属性**按钮。将显示一个窗口。要对索引添加数字属性，单击**添加**按钮，将显示另一个窗口。指定属性的 SQL 列表达式和名称。

或者，选择一种属性，并按适当的按钮来更改、移动或除去条目。

使用“数字”属性不仅可对文本列建立索引，还可对数字列表达式建立索引。例如，如果不仅想要对文本列还要对类型为 `TIMESTAMP` 的列日期建立索引，应指定数字属性 `"cast(julian_day(date) as double)"`，并为该属性指定名称。数字属性的数据类型必须为 `DOUBLE`。

如果想要在搜索查询中使用数字表达式，那么应指定数字属性。

文本属性面板

用途

此面板允许您指定文本文档的语言和格式。如果文档与数据库不是采用相同的 CCSID 存储的，并且文本列属于二进制类型，那么应指定 CCSID。

注意，最初已经选择了数据库 CCSID。如果文档具有 GPP、HTML、Outside In 或 XML 结构化格式，那么可以指定文档模型。

注: 在格式列表框中，Outside In 过滤格式称为 INSO。

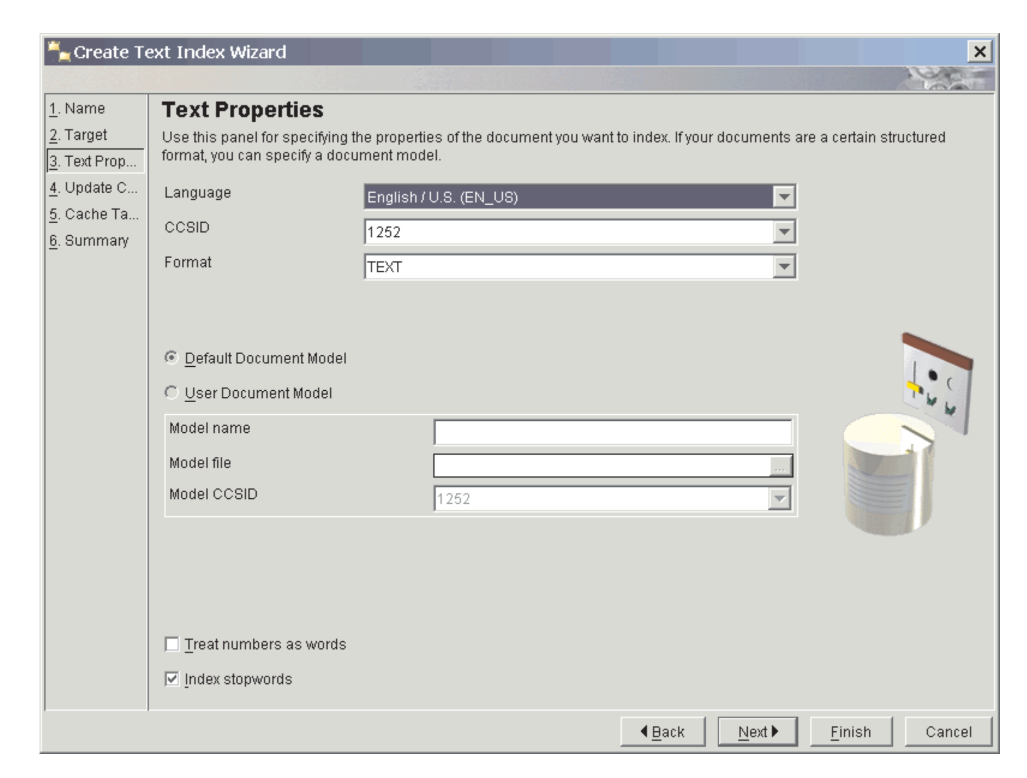


图 13. 创建文本索引向导: 文本属性面板

以下是对该面板中各字段的描述:

表 3. 文本属性面板文本字段

字段名称	必填/可选	缺省值	描述
语言	可选	EN_US	当对文档建立索引时，选择一种语言以确定句子结束定界符和段结束定界符。
CCSID	可选	数据库的 CCSID	选择用于对文本文档建立索引的 CCSID。
格式	可选	如果列类型是 XML，那么采用 TEXT 或 XML	选择文本文档格式: HTML、XML、TEXT、INSO 或 GPP。
缺省文档模型	可选	启用	使用缺省文档模型。
用户文档模型	可选	禁用	使用您的文档模型。

表 3. 文本属性面板文本字段 (续)

字段名称	必填/可选	缺省值	描述
模型名称	如果选择了“用户文档模型”，那么此字段是必填字段	无	输入文档模型的名称。对于 HTML、XML、Outside In 和 GPP 格式，可以指定文档模型。注意，只有在模型文件中才能找到该名称。
模型文件	同上	无	指定文档模型文件。DB2 实例所有者必须可以读取该文件。
模型 CCSID	同上	数据库 CCSID	选择 CCSID 以解释文档模型文件的内容。
将数字作为单词来处理	可选	禁用	选择此项以将一系列数字解释为独立的单词，即使它们与字符相邻也是这样。
索引停用词	可选	启用	选择此项以启用特定于语言的停用词处理。`sqllib/db2ext/resources` 目录中的 `<language>.tsw` 文件包含停用词列表。

更新特征面板

用途

此面板允许您指定索引是以增量方式更新还是从头开始重新创建。可以指定更新设置，以便索引会在指定时间自动更新。

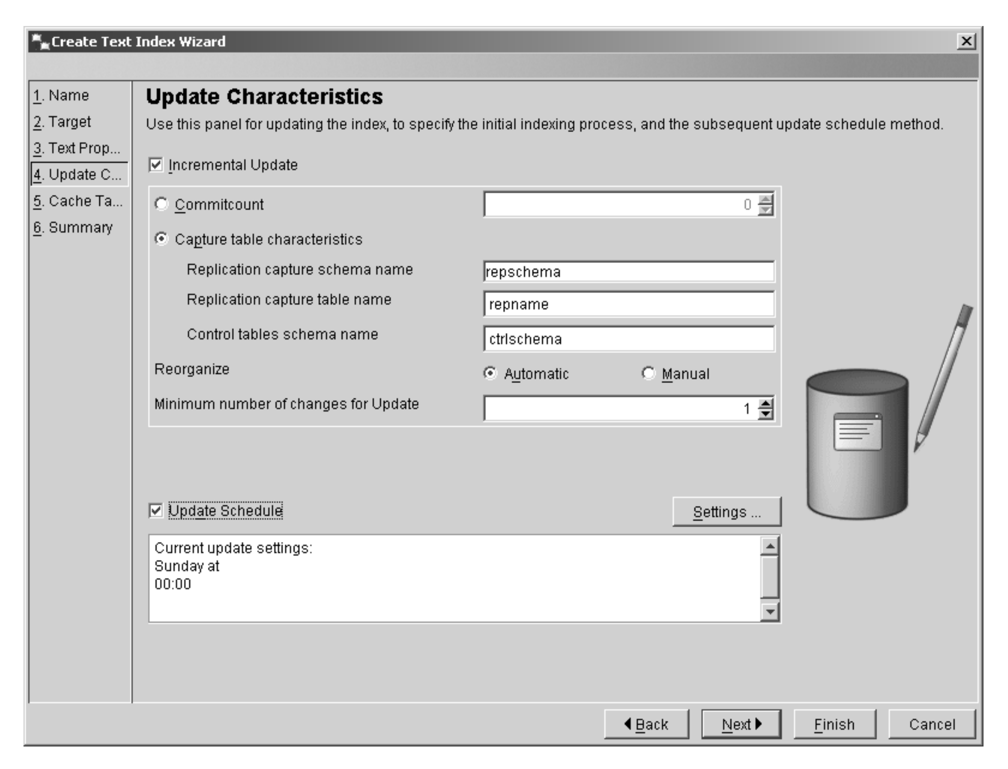


图 14. 创建文本索引向导：更新特征面板

以下是对该面板中各字段的描述:

表 4. 更新特征面板文本字段

字段名称	必填/可选	缺省值	描述
增量更新	可选	启用	选择此项以进行增量索引更新。如果不启用该复选框，那么在执行更新操作时将重新创建索引。
落实计数	可选	0	在一个事务中执行更新期间所处理的更改数。建议不要更改此缺省值。 使用非零落实计数对性能会产生影响。
捕获表特征	可选	无	选择此项以使用复制捕获表来捕获对源表的更改。复制捕获表必须是“捕获数据”（CD）表或“捕获更改数据”（CCD）表，并替换 DB2 Net Search Extender 生成的日志表。
复制捕获模式名	可选	用户标识	复制捕获表的模式名。注意，先前必须已经使用 DB2 复制创建了该表。
复制捕获表名	如果启用了捕获表特征，那么此字段是必填字段	无	复制捕获表的表名。注意，先前必须已经使用 DB2 复制创建了该表。
控制表模式名	如果启用了捕获表特征，那么此字段是必填字段	无	控制表模式名。注意，先前必须已经使用 DB2 复制创建了控制表。
自动或手动重组单选按钮	可选或必填	启用/禁用	自动或手动完成索引重组。
“更新”的最小更改数	可选	1	指定在指定时间对索引进行增量更新之前对文本文档所作的更改的最小数目。
更新调度	可选	禁用	选择此项以添加自动更新设置。

要添加索引更新设置，单击**设置**按钮。注意，仅当您选择了**更新调度**时才会启用此按钮。在此对话框中，选择更新时间的天、小时和分钟。注意，如果选择了很多天，那么会在所选择的每一天的同一时间进行更新。

高速缓存表面板

用途

此面板除了允许您指定索引之外，还允许指定高速缓存的表。可以指定要高速缓存的结果列，并且可以使用存储过程来搜索高速缓存。还可以指定其他高速缓存参数，例如，类型、最大大小以及在初始建立索引期间检索用户表的内容所采用的顺序。

图 15. 创建文本索引向导：高速缓存表面板

以下是对该面板中各字段的描述:

表 5. 结果高速缓存面板文本字段

字段名称	必填/可选	缺省值	描述
启用表高速缓存	可选	禁用	选择此项以启用高速缓存表的构建。
结果列表	如果选择了“启用表高速缓存”，那么此字段是必填字段	无	显示用来指定搜索结果列的 SQL 列表达式的列表。
最大文档数	必填	表的行计数	请参阅下面这一节：确定高速缓存利用率和高速缓存大小。
平均高速缓存行长度	必填	无	请参阅下面这一节：确定高速缓存利用率和高速缓存大小。
初始高速缓存利用率的百分比	可选	50%	选择保留的用于附加文档的高速缓存的百分比。
最大高速缓存大小	可选	无	指定在索引激活期间构建的高速缓存表的最大大小。如果该数目太小，那么将无法激活。
初始搜索结果顺序	可选	禁用	选择此项以定义搜索结果顺序。文档将按照在高速缓存表中相同的建立索引顺序返回。进行增量更新之后就不能确保此顺序了。

表 5. 结果高速缓存面板文本字段（续）

字段名称	必填/可选	缺省值	描述
使用持久高速缓存	可选	启用	在取消激活或系统重新引导之后，此选项将启用执行快速激活。注意，必须指定持久高速缓存的目录路径。如果高速缓存是临时的，那么保持禁用状态。

: 确定高速缓存利用率和高速缓存大小

初始高速缓存利用率的百分比指定要保留给附加文档使用的高速缓存的百分比。最大高速缓存大小指定在激活高速缓存期间要构建的高速缓存表的最大大小。这些选项取决于下列因素:

- 表中的实际文档数。
- 期望的更新数。
- 想要高速缓存的 SQL 表达式的平均大小。

可以输入初始高速缓存利用率的百分比和最大高速缓存大小的建议值。或者，也可以每当在最大文档数或平均高速缓存行长度字段中输入值时就计算这些值。

最大文档数值初始设置为表的行计数。根据文档数和期望的更改数来修改此值。包括所有文档更新、添加和删除。

当将 SQL 表达式添加至高速缓存表列表时，就会根据结果的长度来计算平均高速缓存行长度。由于这是基于表中当前具有的行数，所以可能要花相当长的时间来进行计算。如果您知道此值通常比较小，那么修改该值。

例如，如果一个表具有 10 个条目，列表达式的总计为 100，那么这些值是初始设置的。如果期望最大文档数（包括已删除的文档）为 10000，那么输入此数字。如果您知道列表达式平均起来小于计算的值（例如，`VARCHAR(100)` 和已填写的文本大小 10），那么使用此数字作为平均行大小。

要定义初始搜索结果顺序，单击**定义**按钮。注意，仅当选择了“初始搜索结果顺序”复选框时才会启用此按钮。将出现一个对话框显示指定的所有 SQL 列表达式。要添加结果顺序，单击**添加**按钮，并在对话框中，指定 SQL 结果顺序。

要更改、移动或除去条目，选择表达式并单击适当的按钮。

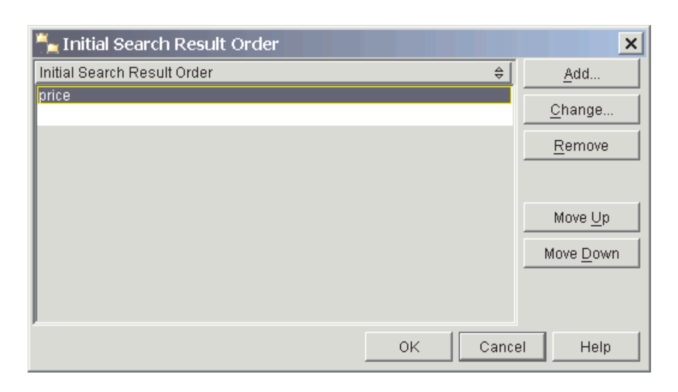


图 16. 初始搜索结果顺序对话框

要添加 SQL 列表达式，单击“结果列”表旁边的**添加**按钮。在该对话框中，指定结果列表达式和名称。

要更改或除去条目，单击将启用适当按钮的列表达式。

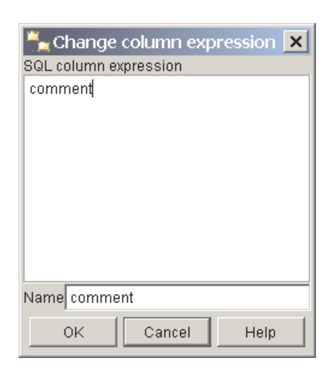


图 17. 更改列表达式对话框

确定高速缓存利用率和高速缓存大小:

初始高速缓存利用率的百分比指定要保留给附加文档使用的高速缓存的百分比。最大高速缓存大小指定在激活高速缓存期间要构建的高速缓存表的最大大小。这些选项取决于下列因素:

- 表中的实际文档数。
- 期望的更新数。
- 想要高速缓存的 SQL 表达式的平均大小。

可以输入初始高速缓存利用率的百分比和最大高速缓存大小的建议值。或者，也可以每当在最大文档数或平均高速缓存行长度字段中输入值时就计算这些值。

最大文档数值初始设置为表的行计数。根据文档数和期望的更改数来修改此值。包括所有文档更新、添加和删除。

当将 SQL 表达式添加至高速缓存表列表时，就会根据结果的长度来计算平均高速缓存行长度。由于这是基于表中当前具有的行数，所以可能要花相当长的时间来进行计算。如果您知道此值通常比较小，那么修改该值。

例如，如果一个表具有 10 个条目，列表达式的总计为 100，那么这些值是初始设置的。如果期望最大文档数（包括已删除的文档）为 10000，那么输入此数字。如果您知道列表达式平均起来小于计算的值（例如，`VARCHAR(100)` 和已填写的文本大小 10），那么使用此数字作为平均行大小。

摘要面板

用途

此面板对先前选择的参数进行了概括。

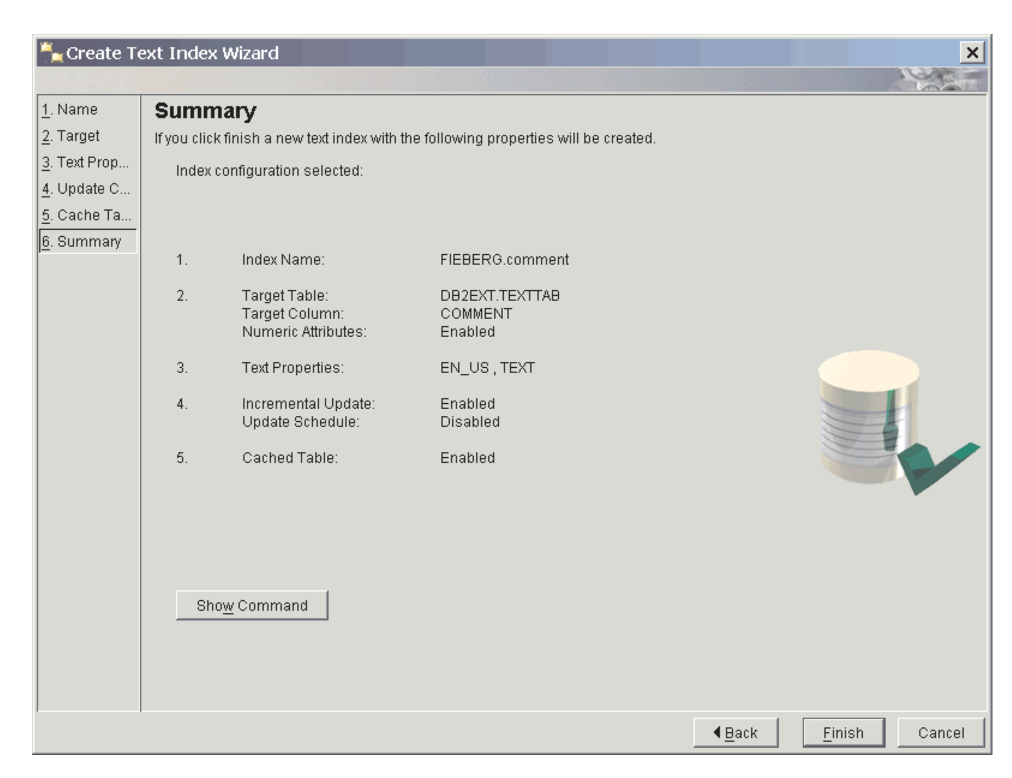


图 18. 创建文本索引向导：摘要面板

单击**显示命令**按钮以查看当单击**完成**按钮时将运行的命令。单击**完成**之后就会创建文本索引。

使用 DB2 控制中心维护文本索引

要维护文本索引，在内容窗格中选择文本索引，并单击**选择**菜单命令。可以从菜单中选择下面的其中一个命令:

1. ALTER 命令，用于改变文本索引。
2. DROP 命令，用于删除文本索引。
3. UPDATE 命令，用于更新文本索引。
4. SHOW INDEX EVENTS 命令，用于显示索引事件。
5. ACTIVATE INDEX MEMORY 命令，用于激活索引高速缓存。
6. DEACTIVATE INDEX MEMORY 命令，用于取消激活索引高速缓存。
7. SHOW STATUS 命令，用于显示索引状态。

注意，仅当使用高速缓存选项创建索引时，才会显示“激活”和“取消激活”命令。

使用 DB2 控制中心改变文本索引

选择**改变**命令，将出现一个对话框显示一系列面板。这些面板对文本索引参数进行了概述。注意，有些参数是不能更改的。

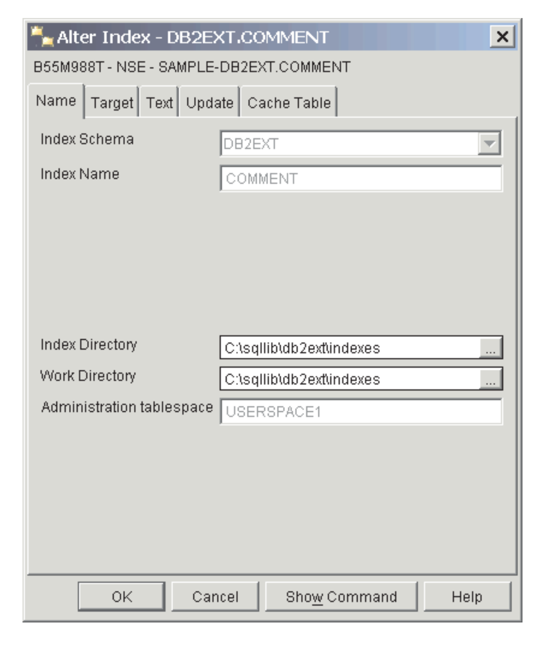


图 19. 改变索引对话框：名称选项卡

名称面板显示索引的名称和存储器配置。可以更改索引目录和工作目录。

目标面板显示索引的目标和数字属性设置。不能更改这些设置。

文本面板显示文本文档配置。不能更改这些设置。

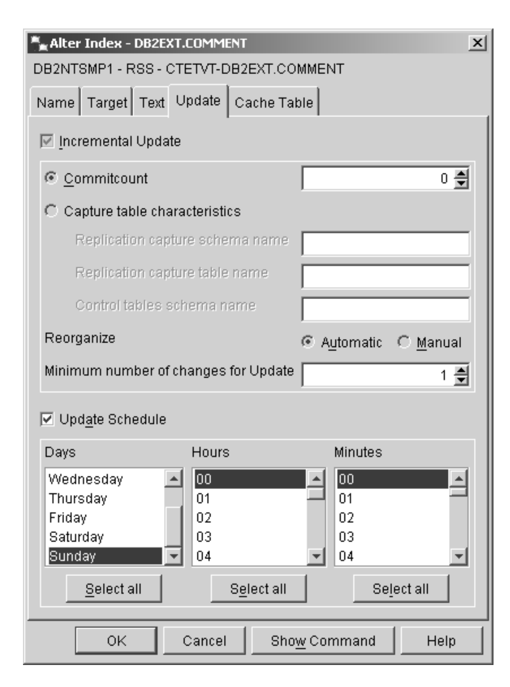


图 20. 改变索引对话框：更新选项卡

更新面板显示索引配置的更新特征。可以改变更新调度。如果索引是为增量更新创建的，那么还可以修改最小更改数。如果索引是使用落实计数选项创建的，那么还可以修改落实计数值。

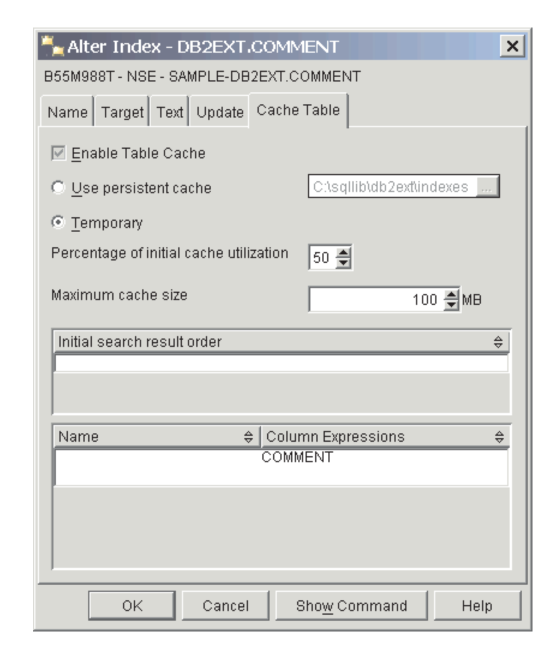


图 21. 改变索引对话框：高速缓存表选项卡

高速缓存表面板显示了高速缓存选项设置。如果已经启用了结果高速缓存，那么可以修改持久目录，也可以使索引高速缓存是临时的。还可以更改"最大高速缓存大小"和"初始高速缓存利用率的百分比"。

使用 DB2 控制中心删除文本索引

选择**删除**命令，将出现一个对话框显示可用的文本索引。

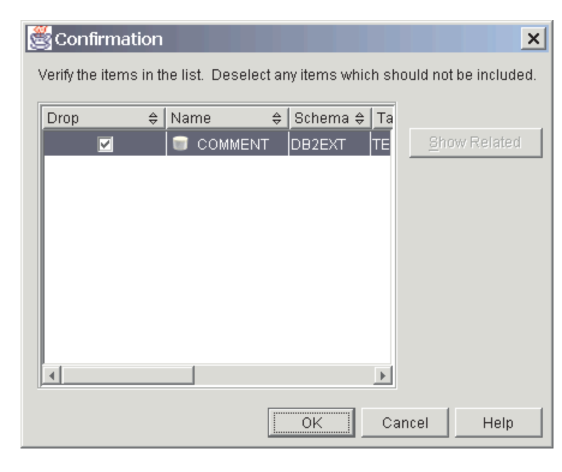


图 22. 删除索引对话框

选择索引，并单击**确定**按钮。

使用 DB2 控制中心更新文本索引

选择**更新**命令，将出现一个对话框显示一些更新选项。

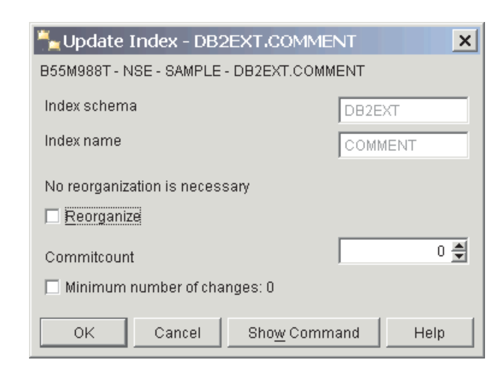


图 23. 更新索引对话框

可以指定更新操作的落实计数。如果想要更新在创建或改变索引期间指定的最小值，那么选择“最小的更改数”复选框。要重组索引，选择该复选框。注意，如果建议进行重组，那么会启用该复选框。

使用 DB2 控制中心显示索引事件

选择**显示索引事件**命令，事件表的内容就会显示在对话框中。将只列示最近的 1000 个事件。

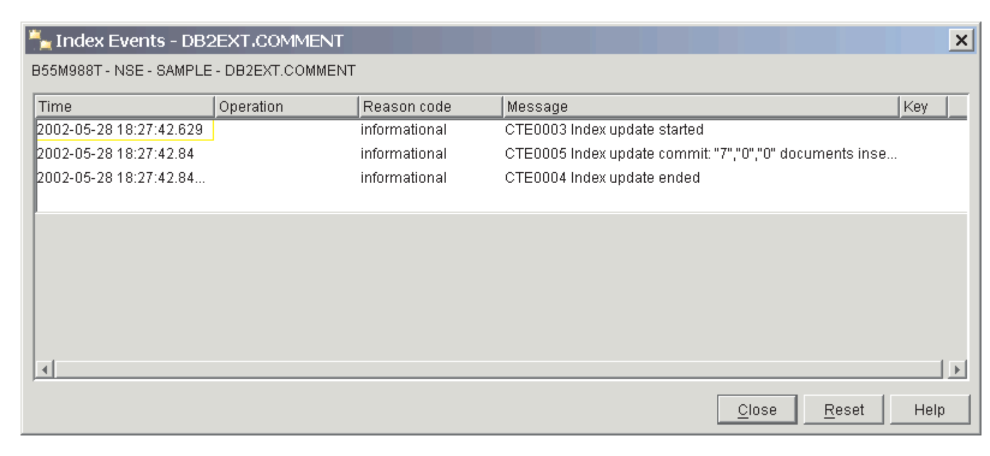


图 24. 索引事件对话框

要清除索引事件，单击**复位**按钮。

使用 DB2 控制中心激活文本索引

选择**激活索引内存**命令，将显示一个对话框。

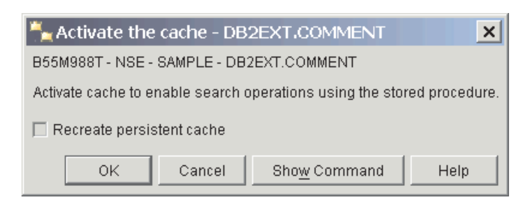


图 25. 激活高速缓存对话框

要激活高速缓存，单击**确定**按钮。如果想要从头开始构建高速缓存，那么选择该复选框。

使用 DB2 控制中心取消激活文本索引高速缓存

选择**取消激活索引内存**命令，将显示一个对话框。



图 26. 取消激活高速缓存表对话框

要释放高速缓存，单击**确定**按钮。

使用 DB2 控制中心显示索引状态

选择**显示状态**命令，将出现一个对话框显示文本索引的状态。

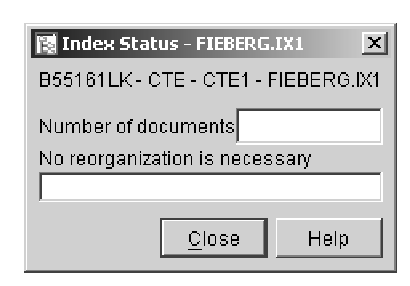


图 27. 索引状态对话框

此对话框包括有关已建立索引的文档数和重组建议标志的信息以及其他索引信息。

第 6 章 开发：创建和维护文本索引

本节提供了有关创建和维护文本索引的信息，它包括下列方面的内容：

- 介绍 `db2text` 命令
- 启用数据库以进行文本搜索
- 创建不同数据类型的文本索引
- 使用增量索引更新（它使用 DB2 复制）来对昵称创建文本索引
- 创建存储过程搜索可以使用的文本索引
- 有关视图的文本索引
- 维护索引

还提供了关于避免可能发生的代码页问题的信息，以及需要考虑的性能注意事项。

在创建文本索引之前，确保已经满足了第 17 页的第 4 章，『规划注意事项』中的先决条件。还要确保已经使用 `db2text start` 命令启动了“Net Search Extender 实例服务”。

注意

还可以使用 DB2 控制中心来创建和维护文本索引。

启用数据库

何时删除文本索引
对包含要搜索的文本列的每个服务器启用一次。

命令 ENABLE DATABASE FOR TEXT

权限 SYSADM

此命令准备已连接的服务器供 Net Search Extender 使用。

此命令还会注册在第 142 页的『SQL 标量搜索函数和 SQL 表值函数』中所描述的 Net Search Extender 搜索函数和过程。

当启用数据库时，此命令还会自动创建下列表和视图：

db2ext.dbdefaults
存储索引、文本和处理特征的数据库缺省值。

db2ext.textindexformats
存储受支持的格式和已使用的当前处于活动状态的模型文件的列表。

db2ext.indexconfiguration
存储索引配置参数。

db2ext.textindexes
用来跟踪所有文本索引的目录视图。

当启用了服务器时，它将保持处于启用状态，直到您禁用它为止。

禁用数据库

何时删除文本索引
当您不再打算在此服务器中进行文本搜索时。

命令 DISABLE DATABASE FOR TEXT

权限 数据库的 DBADM

当 Net Search Extender 准备数据库以供使用时，将进行某些管理更改。本节描述可帮助您取消此过程的一些功能。

要禁用已连接的子系统，使用以下命令:

```
db2text DISABLE DATABASE FOR TEXT
```

当禁用服务器时，该命令将删除下列对象:

- 在启用服务器时创建的 Net Search Extender 目录视图和表。
- Net Search Extender 的 SQL 函数（UDF）的声明。

如果 `DISABLE DATABASE FOR TEXT` 命令返回错误，而您无论如何都想禁用数据库（即使仍在使用索引也想禁用），那么使用以下命令:

```
db2text DISABLE DATABASE for text force
```

注: 如果在数据库中定义了任何文本索引，那么禁用数据库将失败。建议依次除去这些索引，然后检查是否发生任何问题。如果使用 `disable database for text force` 命令，那么它只保证除去数据库中的 Net Search Extender 目录表。

但是，如果不能彻底删除某些索引，那么可能仍然有一些资源需要手动清除。这些资源包括:

- 索引、工作和高速缓存目录中的文件
- ctedem.dat 中的调度程序条目
- 其中索引是使用复制捕获选项创建的，而远程数据库的表中的 IBMSNAP_SIGNAL、IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCNTL 条目必须手动删除。使用 `APPLY_QUAL='NSEDB2'||<instance name> and TARGET_SERVER= <database name>` 条件可以很容易标识这些条目。

  在以下示例中，实例是 DB2，而数据库是 SAMPLE。

  ```
  DELETE FROM <ccSchema>.IBMSNAP_SIGNAL
  WHERE SIGNAL_INPUT_IN IN
          (SELECT MAP_ID FROM <ccSchema>.IBMSNAP_PRUNCNTL
                  WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE');

  DELETE FROM <ccSchema>.IBMSNAP_PRUNCNTL
  WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';

  DELETE FROM <ccschema>.IBMSNAP_PRUNE_SET
  WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';
  ```

创建文本索引

何时删除文本索引
对包含要搜索的文本的每个列创建一次。

命令 CREATE INDEX ... FOR TEXT ...（请参阅下列示例）

权限 对表的 CONTROL 权限

尽管对下列数据类型具有不同的要求，您还是可以对所有数据类型创建文本索引:

- 二进制数据类型
- 不受支持的数据类型

为存储过程搜索创建文本索引也具有不同的要求。

DB2 Net Search Extender 不支持对使用范围分区功能创建的表创建文本索引。这将导致错误“CTE0135: 对象 ″schemaname″.″tablename″ 不存在。

当创建文本索引时，Net Search Extender 会自动创建下列对象:

日志表 此表将跟踪对用户表中的行所作的所有更改。注意，如果选择**更新时重新创建索引**选项或者使用复制捕获表，那么不会创建日志表。

事件表 此表将收集有关在更新文本索引期间的所有更新和潜在问题的信息。

用于用户表的触发器

每当添加、删除或更改用户表中的文档时，这些触发器就会将信息添加至日志表。在下一次按预定时间更新索引或者手动更新索引期间，要保持索引同步就需要该信息。

注意，仅当创建日志表时才会创建触发器，并且，将对基本表而不对视图或昵称表创建文本索引。

要优化性能和磁盘空间，`CREATE INDEX` 命令可以使用一个选项来为表指定另一个表空间。

> **注**: 如果使用 DB2 LOAD 命令来导入文档，那么不会触发触发器，并且不可能对已装入的文档建立增量索引。
>
> 因此，最好是在 DB2 IMPORT 命令激活触发器时才使用该命令。

以下示例将对 `htmltab` 表中的 `HTMLFILE` 文本列创建文本索引。

```
db2text create index DB2EXT.HTMLIDX for text on DB2EXT.HTMLTAB
          (HTMLFILE) format HTML
```

此表上必须存在主键。

用于创建索引的缺省值来自于 `db2ext.dbdefaults` 视图。

如果在建立索引期间发生了错误，那么会将**索引更新事件**行添加至事件表。例如，当找不到为建立索引而排队的文档或者文档格式无效时就会发生这种情况。要获取更多信息，请参阅第 205 页的『事件视图』的描述。

要取消由 `CREATE INDEX` 所作的更改，使用 `DROP INDEX` 命令。有关此信息，请参阅第 54 页的『删除文本索引』。

要使用文本列中的数据来填充所创建的索引，使用以下命令:

```
db2text update index DB2EXT.HTMLIDX for text
```

注意，只有通过使用 db2text update 命令使文本索引与表同步之后才能成功地搜索文档。

注: 搜索摘要

根据在创建索引期间所选择的选项，可以采用不同的搜索方法:

- SQL 标量搜索函数处理所有文本索引（除了对视图创建的文本索引）。
- 存储过程搜索函数只处理使用高速缓存创建的文本索引。
- SQL 表值函数处理所有文本索引（包括对视图创建的文本索引）。

对二进制数据类型创建文本索引

当将数据存储在具有二进制数据类型（例如，BLOB 或 FOR BIT DATA）的列中时，DB2 不会转换数据。这意味着文档将保留它们的原始代码页（CCSID），这在创建文本索引时将导致问题，这是因为您可能具有两个不同的代码页。因此，需要确定您是正在使用数据库的代码页，还是正在使用在 CREATE INDEX 命令中指定的代码页。

为了避免产生此问题，应在创建文本索引时指定代码页:

```
db2text CREATE INDEX db2ext.comment FOR TEXT ON db2ext.texttab (comment)
         CCSID 1252
```

如果没有指定代码页，那么通过调用以下命令来检查已经使用了哪个 CCSID 来创建索引:

```
db2 SELECT ccsid FROM db2ext.textindexes WHERE INDSCHEMA = 'DB2EXT'
                       and INDNAME = 'COMMENT'
```

注意，不支持具有不同代码页的文档位于同一个文本索引中。有关 DB2 如何转换文档代码页设置的信息，请参阅《国际化指南》。

注意，在对字符数据类型创建索引时并不存在该问题。对于字符数据类型，不要指定 CCSID 参数。

对不受支持的数据类型创建文本索引

要创建索引，文本列必须属于下列数据类型的其中一种:

- CHAR
- VARCHAR
- LONG VARCHAR
- CLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- DBCLOB
- BLOB
- XML

如果此列表中并不包含文档所在的列所具有的数据类型（例如，用户定义的类型（UDT）），那么必须提供一个转换函数，它将用户类型作为输入，然后将该用户类型强制转换为一种有效数据类型来作为输出类型。

然后，在创建索引时必须指定此转换函数的名称。有关进一步的信息，请参阅第 114 页的『CREATE INDEX 命令』。

示例：您打算将压缩文本存储在表中。

1. 在交互式 SQL 会话中为文本创建用户定义的类型（UDT）：

```
db2 "CREATE DISTINCT TYPE COMPRESSED_TEXT AS CLOB(1M)"
```

2. 创建一个表，并将文本插入该表中:

```
db2 "CREATE TABLE UDTTABLE (author VARCHAR(50) not null,
                            text COMPRESSED_TEXT, primary key (author))"
db2 "INSERT ..."
```

3. 例如，创建一个称为 uncompress 的用户定义的函数（UDF）。这将接收到一个类型为 COMPRESSED_TEXT 的值，并且返回相应的未压缩文本，例如，值 CLOB (10M)。
4. 按以下方法创建文本索引来指定 uncompress UDF:

```
db2text "CREATE INDEX UDTINDEX for text ON UDTTABLE
                                                  (uncompress(text))
                                   ..."
```

使用增量索引更新（它使用 DB2 复制）来对昵称创建文本索引

在使用复制捕获表对昵称创建文本索引之前，必须执行下列步骤:

1. 使用所有服务器定义和包装器定义设置 DB2 联合数据库。
2. 在远程服务器中设置复制控制表和 Capture 程序。这就是昵称的源表所在的地方。如果 DB2 不自动创建昵称，那么必须使用下列各表的一个模式名以在联合 DB2 数据库中创建昵称:
   - IBMSNAP_SIGNAL
   - IBMSNAP_PRUNE_SET
   - IBMSNAP_PRUNCNTL
   - IBMSNAP_REGISTER
   - IBMSNAP_REG_SYNC（仅适用于非 DB2 远程源）

   在执行此步骤之后，复制控制表的昵称是作为 DB2 联合数据库上的一种“捕获控制模式”中的昵称提供的。此模式名对于 DB2TEXT CREATE INDEX 命令很重要。
3. 将表注册为复制源。
4. 如果在注册步骤中 DB2 未自动创建昵称，那么在联合数据库中为复制捕获表创建昵称。复制捕获表可以是“更改数据”（CD）表，也可以是“一致更改数据”（CCD）表。此昵称是 DB2TEXT CREATE INDEX 命令的一个参数。

   注意，不能更改列名 IBMSNAP_OPERATION、IBMSNAP_COMMITSEQ、IBMSNAP_INTENTSEQ 和主键列的名称。

5. 如果正在使用 DB2 复制源，那么确保 Capture 程序正在运行。不要冷启动 Capture 程序；如果要使用冷启动，那么必须插入 APPLY_QUAL LIKE 'NSE%' 的 IBMSNAP_SIGNAL 表中的所有行。以下 SQL 语句可以让您了解是如何完成此任务的:

```
INSERT INTO <capture control schema>.IBMSNAP_SIGNAL
SELECT CURRENT TIMESTAMP, 'CMD', 'CAPSTART', MAP_ID, 'P'
FROM <capture control schema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL LIKE 'NSE
```

6. 可以使用以下示例，通过使用复制对昵称创建文本索引:

```
DB2TEXT
CREATE INDEX <indexname> FOR TEXT ON <nickname> (< text column>)
REPLICATION CAPTURE TABLE <capture nickname>
CONTROL TABLE SCHEMA <capture control schema>
```

创建存储过程搜索可以使用的文本索引

如果您预先知道想将表中的哪些数据提供给用户，并且您只对位于顶部的搜索结果感兴趣，而不是对整个结果列表感兴趣，那么可以使用存储过程搜索。对于存储过程搜索，需要在执行 CREATE INDEX 命令期间指定高速缓存选项。通过使用已高速缓存的索引，在查询时就可以获得高性能，这是因为它会将指定的所有数据移到主存储器中，从而避免了从表中执行物理读取操作所花费的高成本。

在首次更新已高速缓存的索引之前，应确保表中已包含文档以避免更新未填充表的索引。这样可获得更好的建立索引性能和正确估计高速缓存要求。

存储过程搜索可以快速返回与文档相关联的预定义数据。在 CREATE INDEX 命令中使用高速缓存表选项来进行定义。然后，ACTIVATE CACHE 命令将指定的数据移入内存高速缓存中。

当为存储过程搜索创建文本索引时，必须确定和计算下列参数:

- 高速缓存的类型（临时高速缓存或持久高速缓存）
- 索引更新的类型（自动更新、增量更新或者在每次更新时重新创建索引）
- Net Search Extender 可以使用的最大内存量，使用 MAXIMUM CACHE SIZE 来设置。
- 后续文档更新所需要的可用内存量，使用 PCTFREE 来设置。注意，这只适用于增量更新。

提供了下列类型的高速缓存:

临时高速缓存

这是使用每个 DB2TEXT ACTIVATE CACHE 命令重建的，并且需要将数据从 DB2 表重新装入内存中。如果每次重新启动 Net Search Extender 或者重新引导系统时都从头开始构建已高速缓存的索引，那么花费的时间将比重新激活持久高速缓存的时间更长，对于大型表更是如此。仅当您在处理少量的固定数据，并且无须考虑构建已高速缓存的数据所花的时间时才使用临时高速缓存。

持久高速缓存

持久高速缓存保存在磁盘上，可以使用 DB2TEXT ACTIVATE CACHE 命令将它快速映射至内存。在增量索引更新方案中，高速缓存必须保持激活状态，以使表与已高速缓存的索引保持同步。否则，下一个 DB2TEXT ACTIVATE CACHE 命令将从头开始重新创建高速缓存。

提供了下列方法来更新文本索引:

“更新时不重新创建索引”选项

如果未设置“更新时重新创建索引”选项，那么会进行自动更新索引。由更新索引命令触发此过程，由“更新频率”选项来确定更新时间间隔。此更新过程又称为增量更新。

应避免删除表中的文档然后再重新插入文档，这是因为在高速缓存中不能复用已删除文档的槽。因此，应避免更改已激活的索引的键列。

“更新时重新创建索引”选项

每次更新时，此选项都将重新创建索引。在高速缓存列表达式中的任何可能位置使用 variable 数据类型。这将节省高速缓存空间。在 CACHE TABLE 子句中使用相应的 CAST 表达式。

如果数据不是很稳定，即，如果期望在初始索引激活之后插入超过 50% 的文档，那么使用此选项。

Net Search Extender 提供了两个 SQL 函数来帮助您确定 CREATE INDEX 内存参数。这两个 SQL 函数是 MAXIMUM CACHE SIZE 和 PCTFREE。

- MAXIMUM CACHE SIZE 指定已高速缓存的索引的最大大小。可以使用以下 UDF 函数来获取 MAXIMUM CACHE SIZE 的值（按兆字节（MB）计）。

  ```
  DB2EXT.MAXIMUM_CACHE_SIZE(maximumNumberDocs INTEGER,
              averageRowLength INTEGER, numberOfCacheColumns INTEGER)
  ```

  以下命令将返回表的平均行长度参数:

  ```
  SELECT AVG(LENGTH(cache column_1) + ... + LENGTH(cache column_n))
  ```

  注意，如果将更多文档插入到表中，那么平均值可能会极大地更改。高速缓存列数目与在 DB2TEXT CREATE INDEX 命令的 CACHE TABLE 子句中使用的列表达式数目相关。

  要获取更多信息，请参阅第 17 页的『存储过程搜索的内存要求』。

- PCTFREE 指定 MAXIMUM CACHE SIZE 中所指定的可用来存放更多文档的高速缓存所占的百分比。以下 UDF 函数将根据实际文档数和最大文档数来返回建议的 PCTFREE 值。

  ```
  DB2EXT.PCTFREE(actualNumberDocs INTEGER, maximumNumberDocs INTEGER)
  ```

  实际文档数是第一次执行 ACTIVATE CACHE 命令（该命令将创建内存高速缓存）时表中具有的行数。

  最大文档数是在运行下一个 DB2TEXT ACTIVATE 命令（对于临时高速缓存）或 DB2TEXT ACTIVATE CACHE RECREATE 命令（对于持久高速缓存）之前估计的表中的最大文档数。

  缺省值设置为 50%。如果每次更新时都重新创建索引，那么将 PCTFREE 值设置为 0。

假定您的表有 10000 行，而您不希望超过 20000 行。可使用以下调用来计算需要的 PCTFREE 值:

```
db2  "values DB2EXT.PCTFREE(10000,20000) "
```

假定最大行大小为 20000，且在高速缓存中具有平均大小为 76 的 2 列。使用以下调用来返回该大小:

```
db2 " values DB2EXT.MAXIMUM_CACHE_SIZE(20000,76,2) "
```

在确定相应的参数之后，可以使用以下调用来创建高速缓存的索引:

```
db2text CREATE INDEX db2ext.comment FOR TEXT ON db2ext.texttab (comment)
                 CACHE TABLE (docid) PCTFREE 10 MAXIMUM CACHE SIZE 5
```

在此示例中，对 docid 列进行了高速缓存，并将主存储器用于快速返回结果表。保留 10% 的高速缓存供以后的文档使用，并将高速缓存限制为最多 5 MB。

创建视图的文本索引

可以创建有关视图的文本索引来与存储过程或表值搜索函数配合使用，但是，不能使用任何标量函数，例如，CONTAINS。

另一个主要局限性是不能对视图创建触发器，因此，不能自动识别底层基本表中的任何更改。

因此，对于增量索引更新，用户必须清楚已经添加、更新或删除了哪些文档以便使文本索引与数据库同步。为此，必须将所有更改添加至日志表。以下样本中说明了此过程:

1. 使用以下命令来创建基本表:

   ```
   db2 "create table sample (key INTEGER not null  PRIMARY KEY, name
                       VARCHAR(50) not null, comment VARCHAR(90))"
   ```

2. 使用下列命令来添加一些条目:

   ```
   db2 "insert into sample values(1,'Claus','works in room 301')"
   db2 "insert into sample values(2,'Manja','is in the same office
                                                         as Juergen')"
   db2 "insert into sample values(2,'Juergen','has the longest way to
                                                         Raiko')"
   db2 "insert into sample values(3,'Raiko','is sitting in the office
                                                         besides Claus ')"
   ```

3. 使用以下命令来创建视图:

   ```
   db2 "create view sampleview as select key, comment from sample"
   ```

4. 使用下列命令来创建、更新和激活文本索引:

   ```
   db2text "create index indexview for text on hde.sampleview(comment)
                       cache table (comment) maximum cache size 1 key columns
                       for index on view (key)"
   db2text "update index indexview for text"
   db2text "activate cache for index indexview for text"
   ```

> **注**: 需要指定高速缓存表以便能够对视图创建文本索引。要创建正确的日志表，必须对视图的索引指定键列。如果按此方式创建索引，那么还可以使用表值函数来搜索索引。
>
> 当在分布式 DB2 环境中使用存储过程搜索时，必须在单个节点上为管理表显式指定表空间，并在此节点上显式调用。为了确保连接至正确的节点，使用 DB2NODE 环境变量。

5. 要更新表，使用下列命令:

   ```
   db2 "insert into sample values(4,'Bernhard','is working on the same floor
                       as Manja, but not as Claus')"
   db2 "insert into sample values(5,'Guenter','shares the office with Raiko')"
   ```

6. 然后，更新日志表。要获取日志表的名称，使用以下命令:

```
db2 "select INDSCHEMA,INDNAME,LOGVIEWSCHEMA,LOGVIEWNAME
                 from db2ext.textindexes"
```

以下是日志表的布局:

```
sqltype              sqllen  sqlname.data             sqlname.length
--------------------  ------  -----------------------  --------------
496   INTEGER              4  OPERATION                             9
392   TIMESTAMP           26  TIME                                  4
497   INTEGER              4  PK01                                  4
```

要将条目添加到日志表中，使用下列命令:

```
db2 "insert into sample values(0,CURRENT TIMESTAMP,4)"
db2 "insert into sample values(0,CURRENT TIMESTAMP,5)"
```

第一个值用来描述操作（0 = 插入，1 = 更新，2 = 删除）。第二个值应当始终是 CURRENT TIMESTAMP，而最后一个值是已经插入、更新或删除的行的主键。

7. 使用以下命令再次更新该索引:

```
db2text "update index indexview for text"
```

现在，可以使用存储过程对新值进行搜索了。

建立索引时的性能注意事项

要在建立索引期间提高性能，应考虑下列问题:

- 使用 VARCHAR 数据类型而不是 LONG VARCHAR 或 CLOB 来存储文本文档。
- 使用单独的物理磁盘来存储文本索引和数据库文件。
- 使用较小的主键列，例如，TIMESTAMP 和 INTEGER 而不是 VARCHAR 类型。
- 确保系统具有足够的实内存可用于所有此类数据。如果内存不足，那么操作系统就会使用调页空间。这会降低建立索引和搜索性能。
- 在自动或手动更新索引期间使用的更新 commitcount 参数在增量建立索引期间会降低建立索引的性能。注意，在初始更新过程中未使用该参数。
- 如果将许多错误消息和警告消息写入事件日志表，那么在更新索引期间性能可能会下降。

注: 有关最新的性能提示，请访问以下 DB2 Net Search Extender Web 站点: http://www.ibm.com/software/data/db2/extenders/netsearch/index.html

维护文本索引

本节描述如何维护文本索引和获取有关它们的状态的有用信息。维护任务包括:

1. 更新和重组文本索引
2. 改变文本索引
3. 清除（删除）索引更新事件信息
4. 删除文本索引
5. 查看索引状态

本节还包括有关如何备份和复原索引和已启用的数据库的信息。

更新和重组文本索引

在第一次创建和更新文本索引之后，必须保持文本索引是最新的。例如，当将文本文档添加至表或者更改表中的现有文档时，必须对该文档建立索引以使索引内容与该表的内容同步。同样，当从表中删除文本文档时，必须从索引中除去其项引用。

如果未使用 RECREATE INDEX ON UPDATE 选项来创建文本索引，那么触发器会自动将有关新文档、已更改文档或已删除文档的信息存储在内部日志表中。在下一次更新索引时，将对日志表中引用的文档建立索引。

如果在 CREATE INDEX 命令中指定了 RECREATE 选项，那么每次更新时都会彻底重建索引。此选项不会创建任何日志表或触发器。如果您具有大型表，由于重建完整的索引需要很高的成本，因此应谨慎使用此选项。

通常，按给定的时间间隔来自动更新索引。可以使用 ALTER INDEX 命令来更改现有索引的更新频率。

根据要进行更新的时间以及在开始更新索引之前必须在日志表中进行排队的最小文本更改数来指定索引的更新频率。如果在给定的日期和时间内，日志表中没有足够的更改，那么不会更新索引。

应当仔细地计划定期建立索引；对大量文本文档建立索引是一项既耗费时间又耗费资源的任务。它所花的事件取决于很多因素。这些因素包括：文档大小、自从上一次更新索引以来已经添加或更改了的文本文档数以及处理器的能力。

应当避免对大量索引进行频率非常高的自动更新，这样做可能会导致产生死锁。例如，如果对 100 个索引每隔 5 分钟就更新一次，并且每周 7 天且每天 24 小时不间断地执行此操作，那么一周就会生成一个包含 100*12*24*7=201600 个检查点的内部列表，管理此列表的成本是很高的。

注: 在 DB2 表上，在下列情况下可能会发生回滚和死锁情况:

- 高更新频率
- 高频率更改事务
- 长事务

更新了数据库表之后，就会将必须对 Net Search Extender 索引进行的更改记录到日志表中。在处理这些日志表条目之后，就会从日志表中删除这些条目。如果对日志表执行的这些删除操作与需要记录的数据库表中的更新一致，那么可能会发生死锁情况。

更新文本索引

UPDATE INDEX 命令允许您一旦请求就立即更新索引。

何时删除文本索引

当必须立即更新索引而不等待定期建立索引时。

命令 UPDATE INDEX

权限 对表的 CONTROL 权限

以下命令将更新索引:

```
db2text UPDATE INDEX comment FOR TEXT
```

当已将几个文本文档添加至数据库并且想要立即搜索它们时，此命令很有用。

如果在执行 CREATE INDEX 期间指定了 AUTOMATIC REORGANIZE，那么在需要时将自动重组索引。

如果您指定 MANUAL REORGANIZATION 并且想确定是否需要进行手动重组，那么使用以下命令来查询 db2ext.textindexes 视图：

```
db2 "select reorg_suggested from db2ext.textindexes where INDNAME = 'comment'"
```

如果您指定 MANUAL REORGANIZATION 并且经常更新一列，那么应记住更新过程将变慢。要手动进行重组，使用以下命令：

```
db2text UPDATE INDEX comment FOR TEXT reorganize
```

改变文本索引

何时删除文本索引
当必须更改更新频率或者索引目录和工作目录时。

命令 ALTER INDEX

权限 对表的 CONTROL 权限

使用此命令来更改索引工作目录、索引的更新频率或者高速缓存特征（主要是 MAXIMUM CACHE SIZE 或 PCTFREE）。如果不指定更新频率，那么当前设置将保持不变。如果正在更新或搜索索引，那么会显示错误消息。这表明当前锁定了索引，不能执行任何更改。

以下示例将更改索引的更新频率。

```
db2text ALTER INDEX comment FOR TEXT
         UPDATE FREQUENCY d(1,2,3,4,5) h(12,15) m(00) UPDATE MINIMUM 100
```

在此示例中，如果队列中至少有 100 个文本文档，那么将在星期一至星期五的 12:00 或 15:00 更新索引。

使用以下命令来停止索引的定期更新：

```
db2text ALTER INDEX comment FOR TEXT
                                             UPDATE FREQUENCY NONE
```

如果使用 ALTER INDEX 命令来更改索引目录，那么会将索引文件从一个目录复制到另一个目录，并且在此过程中将锁定索引。在复制过程完成之后，就会对索引解锁，于是又可以使用索引了。

清除索引事件

何时删除文本索引
当不再需要索引的事件表中的消息时。

命令 CLEAR EVENTS FOR INDEX

权限 对表的 CONTROL 权限

有关建立索引事件的信息（例如，更新的开始时间和结束时间、已建立索引的文档数或者更新期间发生的文档错误）都存储在索引的事件表中。这可以帮助您确定问题的原因。当不再需要这些消息时，可以删除。

以下示例将从指定的文本索引中删除消息:

```
db2text CLEAR EVENTS FOR INDEX comment FOR TEXT
```

删除文本索引

何时删除文本索引
当您打算不再从文本列中搜索文本时。

命令 DROP INDEX FOR TEXT

权限 对表的 CONTROL 权限

示例:

```
db2text DROP INDEX comment FOR TEXT
```

当删除文本索引时，也删除了下列表和视图:

- 索引的日志表和视图
- 索引的事件表和视图
- 日志表触发器（如果存在）

注: 始终要在删除表的索引之后才废弃表。如果首先废弃表，那么索引仍然会存在。

查看文本索引状态

要获取有关数据库中的当前文本索引的信息，请使用 Net Search Extender 目录视图。例如，如果想知道当前的数据库缺省值，使用以下命令:

```
db2 "select * from db2ext.dbdefaults"
```

要获取当前可用的索引、它们相应的表和已建立索引的文档数的信息，请使用以下命令:

```
db2 "select indschema, indname, tabschema, tabname, number_docs
         from db2ext.textindexes"
```

使用以下命令来获取有关特定索引的格式的信息:

```
db2 "select format, modelname from db2ext.textindexformats where
          indschema = 'DB2EXT' and indname = 'TITLE'"
```

如果未设置 COMMITCOUNT，那么在运行更新过程期间不会更新 db2ext.textindexes 中的 NUMBER_DOCS 参数。要在更新过程中查看当前已更新的文档数，使用以下命令:

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE sample INDEX db2ext.title
```

备份与复原索引

执行下列步骤来**备份**已启用的数据库和由 Net Search Extender 创建的文本索引:

1. 要了解 Net Search Extender 已经创建了哪些索引以及将这些索引存储在何处，可对 db2ext.textindexes 视图调用 Select 语句:
   ```
   db2 "select indschema, indname, indexdirectory from db2ext.textindexes"
   ```
2. 确保未在更新任何索引，然后使用以下命令来停止 Net Search Extender 服务:
   ```
   db2text stop
   ```
3. 在备份数据库之后，备份索引目录和子目录。

4. 使用以下命令重新启动 Net Search Extender 服务:

```
db2text start
```

执行下列步骤来**复原**已启用的数据库和由 Net Search Extender创建的文本索引:

1. 使用以下命令来停止 Net Search Extender:

```
db2text stop
```

2. 将索引目录的备份副本复原到以前的同一路径中。
3. 使用以下命令重新启动 Net Search Extender:

```
db2text start
```

除去 /tmp 目录中的文件

当 Net Search Extender 服务正在运行时，下列文件必须存在于 /tmp 目录中并且不能删除:

- 信号量和共享内存文件:

```
<instance_owner>TEXT.0000.LATCH
<instance_owner>TEXT.0000
<instance_owner>CACHE.0000
<instance_owner>SCHEDULER.LATCH
<instance_owner>DEMON.SEM
<instance_owner>DEMON:MEM
```

- 在创建索引期间，如果高速缓存是临时高速缓存，那么在 /tmp 目录中可能会看到类似于以下内容的文件:

```
<database_name>.IX123456
<database_name>.IX123456.data0
```

第 7 章 搜索文本的方法

Net Search Extender 提供了下列方法来搜索文本:

SQL 标量搜索函数

可以将文本搜索子查询嵌入到 SQL 查询中。Net Search Extender 提供了 SQL 标量搜索函数，这是对可用 SQL 函数的扩展。通过将文本搜索子查询包含在 SQL 查询中，就可以将 Net Search Extender 搜索功能与 DB2 XQuery 处理结合起来。可以在 `db2-fn:sqlquery()` XQuery 输入函数中使用对 XML 文档的文本搜索查询，并且允许使用 XQuery 来直接处理获得的 XML 文档。

存储过程搜索函数

这使您能够返回预定义的高速缓存结果表。

SQL 表值函数

可以采用与存储过程搜索相似的方式来使用此搜索。

对于 SQL 标量搜索函数，本节描述了下列方面:

- 使用 CONTAINS、NUMBEROFMATCHES 和 SCORE 函数来搜索文本。

  有关语法的详细描述，请参阅第 142 页的『SQL 标量搜索函数和 SQL 表值函数』。
- 通过将示例与 CONTAINS 函数配合使用来指定搜索自变量。

  有关语法的完整描述，请参阅第 135 页的『搜索自变量的语法』。

对于存储过程搜索函数，本节描述了下列方面:

- 使用存储过程搜索来搜索文本。
- 要指定搜索自变量，请参阅第 135 页的『搜索自变量的语法』以获取对参数的描述。

对于“SQL 表值函数”，本节描述了下列方面:

- 使用“SQL 表值函数”和 HIGHLIGHT 函数来搜索文本。

  有关语法的描述，请参阅第 142 页的『SQL 标量搜索函数和 SQL 表值函数』。
- 要指定搜索自变量，请参阅第 135 页的『搜索自变量的语法』以获取对参数的描述。

还有一些您可能需要考虑的搜索性能注意事项的信息。

在进行搜索之前，确保已经执行了第 43 页的第 6 章，『开发: 创建和维护文本索引』中描述的所有相应的建立索引步骤，其中包括对不同数据类型建立索引。

使用 SQL 标量搜索函数搜索文本

通过使用一些示例，本节描述了如何按下列方式使用 SQL 标量搜索函数:

- 使用 `CONTAINS` 函数来发出查询。
- 使用 `NUMBEROFMATCHES` 函数来确定在文本文档中找到了多少个相匹配的搜索词条。
- 使用 `SCORE` 函数来获取找到的文本文档的关联性。

有关语法的描述，请参阅第 142 页的『SQL 标量搜索函数和 SQL 表值函数』。

发出查询

此示例演示 CONTAINS 函数如何搜索表 texttab 中的 comment 列中的文本。如果文本满足搜索自变量，那么该函数将返回 1，否则将返回 0。

```
SELECT AUTHOR,TITLE
                FROM DB2EXT.TEXTTAB
        WHERE CONTAINS(COMMENT, '"book"') = 1
```

在此示例中，在 COMMENT 列中搜索词条 book。

不支持搜索 ""。在搜索词条中使用连续引号将导致产生语法错误消息。如果在搜索字符串中使用换行符，那么也会导致查询语法错误错误。

注:

如果您知道文本搜索将返回非常大的结果集，那么添加限制性搜索条件将很有用，例如:

```
SELECT AUTHOR,TITLE
                FROM db2ext.texttab
        WHERE CONTAINS(COMMENT, '"book"') = 1 AND PRICE < 20
```

搜索并返回找到的匹配项数

使用 NUMBEROFMATCHES 函数来确定在每个文本文档中找到搜索词条的频繁程度。

```
SELECT AUTHOR,TITLE,NUMBEROFMATCHES(COMMENT,'"book"')
        FROM DB2EXT.TEXTTAB
    NUMBEROFMATCHES(COMMENT, '"book"') > 0
```

NUMBEROFMATCHES 对于每一行都将返回一个整数值。

搜索并返回找到的文本文档的分数

SCORE 将返回一个正数，它指示一个文档相对于在同一索引中找到的其他文档满足搜索词条的程度。该值是根据在文档中找到的匹配数与文档大小的比例来计算的。在以下示例中，可以使用 SCORE 函数来获取找到的文档的值:

```
WITH TEMPTABLE(docid,score)
                AS (SELECT docid,
                                      SCORE(COMMENT,'"book"')
                FROM DB2EXT.TEXTTAB)
SELECT *
                FROM TEMPTABLE
        WHERE score > 0
                ORDER BY score ASC
```

SCORE 将返回 0 到 1 之间的 DOUBLE 值。

由 SCORE 返回的值只有在与为同一索引返回的其他 SCORE 值进行比较时才有意义。不能将这些值与为其他索引返回的 SCORE 值进行比较。

注: 不能将 CONTAINS、SCORE 和 NUMBEROFMATCHES 搜索函数用于对视图创建的索引。

DB2 环境不同，SCORE 值就不同:

- 在非分布式环境中，所有文档都在单个表中。SCORE 值基于单个表以及文档与表中所有其他文档之间的关系。
- 在分布式 DB2 环境中，所有文档都位于不同的节点上。在建立索引期间，将只使用每个节点上的本地文档来构建文本索引。在这种情况下，SCORE 值基于文档与多个节点的仅其中一个节点中的所有文档之间的关系。

指定 SQL 搜索自变量

CONTAINS、NUMBEROFMATCHES 和 SCORE 函数都使用搜索自变量。本节使用 CONTAINS 函数来说明 Net Search Extender 函数中的搜索自变量的不同示例。

有关语法的完整描述，请参阅第 135 页的『搜索自变量语法』。

按任意顺序搜索词条

一个搜索自变量中可以具有多个词条。组合几个搜索词条的一种方法是使用逗号将它们连接起来，如下所示:

```
SELECT AUTHOR,TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT,
          '("kid", "dinosaur")') = 1
```

这种格式的搜索自变量可以按任意顺序查找包含任何搜索词条的文本。在逻辑词条中，隐式 OR 运算符可连接搜索词条。

使用布尔运算符 AND 和 OR 进行搜索

可以使用布尔运算符“&”（AND）和“|”（OR）来将搜索词条与其他搜索词条组合在一起:

```
SELECT AUTHOR, TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT,
           '"author" | "pulitzer"') = 1
```

还可以使用布尔运算符来组合几个词条:

```
SELECT AUTHOR, TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT,
          '"author" | "pulitzer" & "book"') = 1
```

如果使用多个布尔运算符，那么将按照从左到右的顺序对这些运算符进行求值。但是，与常规布尔逻辑一样，逻辑运算符 AND（&）优先于逻辑运算符 OR（|）。可以在以下示例中了解此求值，它不包括圆括号:

```
"book" & "pulitzer"| "year" & "author"
```

Net Search Extender 按以下方式对布尔运算符进行求值:

```
("book" & "pulitzer") | ("year" & "author")
```

如果想对布尔运算符强制采用其他求值顺序，那么必须添加圆括号:

```
"book" & ("pulitzer" | "year") & "author"
```

还可以将布尔运算符与使用逗号分隔符链接在一起的搜索词条进行组合。

```
("author", "pulitzer") & "book"
```

在这种情况下，逗号将被解释为布尔运算符 OR:

```
("author"| "pulitzer") & "book"
```

使用布尔运算符 NOT 进行搜索

可以使用布尔运算符 NOT 来从搜索中排除特定文本文档:

```
SELECT AUTHOR, TITLE
               FROM DB2EXT.TEXTTAB
               WHERE CONTAINS(COMMENT,
              '("author", "pulitzer") & NOT "book"') = 1
```

此示例将从对“author”或“pulitzer”的搜索结果中排除包含词条“book”的任何文本文档。

模糊搜索

“模糊”搜索将搜索与搜索词条拼写相似的单词。

```
SELECT AUTHOR, TITLE
               FROM DB2EXT.TEXTTAB
               WHERE CONTAINS(COMMENT,
          'fuzzy form of 80 "pullitzer"') =1
```

在此示例中，搜索可能会找到拼写错误的单词 pulitzer。

匹配级别（在该示例中为“80”）指定需要的准确程度。当文档中可能具有拼写错误时，就使用模糊搜索。当“光学字符识别”设备或语音输入创建文档时，通常就会出现这种情况。使用 1 到 100 之间的值来表示模糊匹配的程度，100 表示精确匹配，而低于 80 的任何值都表示“模糊匹配”，值越小，匹配程度就越低。

注: 如果模糊搜索未提供适当的准确度，那么使用字符屏蔽来搜索一个词条的某些部分。

搜索部分词条（字符掩码）

屏蔽字符（或者称为“通配”字符）提供了一种更随意的搜索方法。它们增加了通过搜索找到的文本文档数。

Net Search Extender 使用两个屏蔽字符：百分号（%）和下划线（_）。Net Search Extender 使用这些屏蔽字符与 DB2 谓词 LIKE 使用这些屏蔽字符的方式相同。

- % 表示**任何数目的任意字符**。以下是在搜索词条中部将 % 用作屏蔽字符的一个示例:

  ```
  SELECT AUTHOR,TITLE
                 FROM DB2EXT.TEXTTAB
         WHERE CONTAINS(COMMENT, '"thr%er"') = 1
  ```

  此搜索词条将查找包含单词“thriller”、“throttle”和“thread-splitter”的文本文档。
- _ 表示搜索词条中的**一个字符**。以下示例还将查找包含单词“thriller”的文本文档。

```
SELECT AUTHOR, TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT, '"th_iller"') = 1
```

可以在一个短语中使用多个通配符（短语中的多个单词可以包含一个通配符），但是，通过通配符扩展获得的词条只能是单个词条，而不是由多个单词组成的词条。例如，通配符表达式“thr%er”将不会与短语“the caller”相匹配。

要有限度地使用通配符，这是因为它们会极大地增大结果列表的大小，从而降低性能并且返回不需要的搜索结果。

搜索包含掩码字符的词条

如果想要搜索包含“%”字符或“_”字符的词条，必须在该字符前添加通常所说的转义字符。必须在查询中使用 ESCAPE 关键字来标识转义字符。

在以下示例中，转义字符为 "!":

```
SELECT AUTHOR, TITLE
               FROM DB2EXT.TEXTTAB
               WHERE CONTAINS(COMMENT,
                                '"100!%" ESCAPE "!"') = 1
```

按固定顺序搜索词条

如果您搜索“primary key”，那么仅当这两个词条相邻并且按如下所示的顺序显示时才能够找到它们:

```
SELECT AUTHOR,TITLE
           FROM DB2EXT.TEXTTAB
        WHERE CONTAINS(COMMENT, '"primary key"') =1
```

搜索同一句子或段中的词条

以下是用来查找满足以下条件的文本文档的搜索自变量示例：搜索词条“web”与词条“disk”在同一个句子中。

```
SELECT AUTHOR,TITLE
           FROM DB2EXT.TEXTTAB
           WHERE CONTAINS(COMMENT,
    '"web" IN SAME SENTENCE AS "disk"') = 1
```

还可以搜索同时出现的若干个单词。在以下示例中，将搜索出现在同一段落中的两个短语:

```
SELECT AUTHOR, TITLE
               FROM DB2EXT.TEXTTAB
               WHERE CONTAINS(COMMENT,
      '"linguistic analysis processing" IN SAME PARAGRAPH AS
             "search algorithms"') = 1
```

搜索结构化文档部分中的词条

以下是用来查找满足以下条件的文本文档的搜索自变量示例：搜索词条“IBM”出现在结构化文档的“H2”子节中。

```
SELECT CATEGORY, DATE
                FROM DB2EXT.HTMLTAB
                WHERE CONTAINS(HTMLFILE,
               'SECTIONS ("H2") "IBM"') = 1
```

注意，节名称是区分大小写的。确保模型文件和查询中的节名称是完全相同的。

同义字搜索

同义字搜索是 Net Search Extender 中的一项功能强大的搜索词条扩展功能。搜索的附加词条来自于您自己构建的同义字，因此，您可以直接控制词条。例如，如果您认为“database”与诸如“repository”和“DB2”的词条相关，那么对“database”进行同义字搜索时就可以找到诸如“repository”和“DB2”的词条。

对于您感兴趣的频繁搜索的特定区域使用这种类型的搜索，可以更有效地生成搜索结果。

下列示例将演示使用同义字扩展的语法。

此示例采用词条“product”并通过添加在同义字“nsesamplethes”中找到的此词条的所有相关词条来将它扩展。

```
SELECT CATEGORY, DATE
                FROM DB2EXT.HTMLTAB
                WHERE CONTAINS(HTMLFILE,
           'THESAURUS "nsesamplethes"
                      EXPAND RELATED
           TERM OF "product"') = 1
```

下一个示例采用搜索词条“product”。然后，使用该搜索词条的所有*同义词*来扩展搜索。

```
SELECT CATEGORY, DATE
                FROM DB2EXT.HTMLTAB
                WHERE CONTAINS(HTMLFILE,
           'THESAURUS "nsesamplethes"
                      EXPAND SYNONYM
           TERM OF "product"') = 1
```

数字属性搜索

对于存储在文本索引中的数字属性，可以使用以下语法来进行搜索:

```
SELECT AUTHOR, TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT,
                'ATTRIBUTE "PRICE" between 9 and 20') = 1
```

自由文本搜索

“自由文本搜索”是一种将搜索词条表示为自由格式文本的搜索。短语或句子用自然语言描述要搜索的主题。自由文本查询中单词的顺序是不相关的。但是，在要搜索的文档中必须至少出现自由文本查询中的其中一个查询词条。

注意，不支持屏蔽掉自由文本自变量中的搜索字符串的字符或单词。

例如:

```
SELECT AUTHOR, TITLE, SCORE(COMMENT,
  'IS ABOUT EN_US "something related to dinosaur"')
         FROM DB2EXT.TEXTTAB
         WHERE CONTAINS(COMMENT,
  'IS ABOUT EN_US "something related to dinosaur"') = 1
```

附加搜索语法示例

要熟悉更多搜索语法示例，请运行 search 脚本。这包含针对样本表运行的 Net Search Extender 搜索函数的示例。

按如下所示输入命令:

```
db2 -tvf search
```

如果尚未创建表和索引，那么运行下列其中一项:

- 在 UNIX 平台上: <instance_owner_home>/sqllib/samples/extenders/db2ext 目录中的 nsesample。
- 在 Windows 平台上: <sqllib>/samples/extenders/db2ext 目录中的 nsesample (.bat)。

使用存储过程搜索来搜索文本

如果您只需要文本搜索结果的已排序的子集，并且要求较高的查询性能，那么使用存储过程搜索界面。如果需要所有搜索结果或者需要对大量文档建立索引，那么不要使用存储过程。这主要是因为已将用户表的多个部分复制到内存中，从而需要提供许多实内存。

可以采用一种与游标导航相似的方式使用存储过程来首先请求 0 到 20 之间的结果，然后请求 21 到 40 的结果，依此类推。在具有此游标功能的情况下使用高速缓存（已在建立索引期间计算）会使搜索速度相当快，主要是因为不需要与用户表进行连接。

如果将使用存储过程，那么务必考虑下列选项:

- 在分布式 DB2 环境中，必须在单个节点上为存储过程显式指定表空间，并在此节点上显式调用过程。
- 在 CREATE INDEX 期间已经指定了高速缓存搜索结果选项。
- 已经完全考虑了现在和将来的共享内存要求（可能涉及到增量更新）。
- 已经使用 db2text activate 命令激活了索引的高速缓存。

以下是存储过程搜索的一个示例:

```
db2 "call  db2ext.textSearch('\"book\"','DB2EXT','COMMENT',0,2,1,1,?,?)"
```

第一个参数是搜索词条。搜索词条的语法与 SQL 标量函数中的语法相同。其余参数是索引模式和索引名。如果尚未屏蔽该名称，那么将它转换为大写。接下来的两个数字自变量提供了结果片起始点和该结果片中的结果数。接下来的两个整数值指定是否请求了分数和命中信息。最后两个值是函数返回值。

注: 如果请求较大的结果集，那么需要用户表空间。如果没有提供临时表空间，那么创建一个表空间。以下示例将在 UNIX 平台上创建一个表空间:

```
db2 "create user temporary tablespace tempts managed by system
                   using ('/work/tempts.ts')"
```

在分布式 DB2 环境中，必须在存储过程的单个节点上为管理表显式指定表空间，并在此节点上显式调用过程。

使用 SQL 表值函数搜索文本

如果您不需要所有搜索结果，并且没有足够内存来像存储过程搜索中那样使用高速缓存的索引，那么使用 SQL 表值函数。

提供了两个 SQL 表值函数，它们都称为 db2ext.textsearch。一个函数具有与 db2ext.highlight 函数配合使用的附加参数。

“SQL 表值函数”提供了与存储过程相同的游标界面来仅访问部分结果。但是，仍然需要将结果与用户表连接起来。可以在以下示例中了解这种情况:

```
db2 "select docid , author, score from TABLE(db2ext.textsearch('\"book\" ',
          'DB2EXT','COMMENT',3,2,cast(NULL as integer))) as t, db2ext.texttab u
            where u.docid = t.primkey"
```

下面是一些可能从“SQL 表值函数”返回的值:

```
--> primKey <single primary key type>
the primary key

-->  score           DOUBLE
the score value of the found document

--> NbResults       INTEGER
the total number of found results (same value for all rows)

--> numberOfMatches  INTEGER
the number of hits in the document
```

注: 注意，只允许存在单个主键列。

使用突出显示函数

要使用 SQL 表值函数 db2ext.highlight，必须使用带有附加 numberOfHits 和 hitInformation 参数的 db2ext.textsearch 函数。

在此示例中，会调用 db2ext.highlight 函数来显示整个文档，且不会突出显示由 db2ext.textsearch 函数找到的任何匹配项。

```
select  p.docid,
     db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 0 ')
                  as highlight
from DB2EXT.TEXTTAB p,
     table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
               | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
                  cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

查询将返回以下结果:

```
DOCID HIGHLIGHT

2     A New York Times bestseller about peacekeeping soldiers called
            "Keepers" who devise a shocking scheme to get the worlds
            attention after their tour of duty ends.

1 record(s) selected.
```

在此示例中，会调用 db2ext.highlight 函数来显示整个文档，并且突出显示由 db2ext.textsearch 函数找到的所有匹配项。

```
select  p.docid,
     db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 0,
          TAGS = ("<bf>", "</bf>" ) ') as highlight
from DB2EXT.TEXTTAB p,
     table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
               | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
                    cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

搜索自变量将返回以下结果:

```
DOCID HIGHLIGHT

2     A New York Times <bf>bestseller</bf> about <bf>peacekeeping</bf>
     <bf>soldiers</bf> called "Keepers" who devise a shocking scheme to
      get the worlds <bf>attention</bf> after their tour of duty ends.

1 record(s) selected.
```

在此示例中，会调用 db2ext.highlight 函数来最多显示文档的 10 个部分（窗口）。每个窗口大小为 24 个字符，在匹配项的每一边大约具有 12 个字节的数据。此外，将突出显示由 db2ext.textsearch 表函数找到的匹配项。

```
select  p.docid,
     db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 10,
          WINDOW_SIZE = 24, TAGS = ("<bf>", "</bf>" ) ') as highlight
from DB2EXT.TEXTTAB p,
     table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
               | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
                    cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

搜索自变量将返回以下结果:

```
DOCID HIGHLIGHT

2     York Times <bf>bestseller</bf> about <bf>peacekeeping</bf> ...
      <bf>peacekeeping</bf> <bf>soldiers</bf> called "Keepers" ... the
      worlds <bf>attention</bf> after their

1 record(s) selected.
```

找到的第一个匹配项为 <bf>bestseller</bf>，此匹配项将确定第一个窗口。第二个匹配项 <bf>peacekeeping</bf> 与第一个匹配项只相隔 8 个字节，会将它完全放置在第一个窗口中。第三个匹配项 <bf>soldiers</bf> 在第一个窗口外部，它将确定新窗口。由于第二个匹配项 <bf>peacekeeping</bf> 只与 <bf>soldiers</bf> 匹配项的左边相隔 2 个字节，因此，也会将它放置在第二个窗口中，并进行突出显示。第四个匹配项 <bf>attention</bf> 在第二个窗口外部，所以它将确定新窗口。由于此窗口中没有包含任何先前的或附加的匹配项，因此，该窗口中只包含匹配项附近的数据。

另外，因为没有指定任何 WINDOW_SEPARATOR，所以采用缺省窗口分隔符“...”来分隔这三个匹配项窗口。

注： 为了确保在使用 db2ext.highlight 函数时具有高性能，用户应限制 db2ext.textsearch 表值函数中的搜索结果。

搜索多列

在需要对多列创建文本索引的情况下，最简单的方法是使用 SQL 标量函数并且组合针对这些列的搜索。可在以下示例中看到这一情况：

```
SELECT AUTHOR,TITLE
                FROM DB2EXT.TEXTTAB
                WHERE CONTAINS(COMMENT,
                '"book"')=1 and CONTAINS(AUTHOR,'"Mike"')=1
```

对于表值函数，这会更困难，原因是由于性能原因您可能需要使用返回表的并集。使用表值函数的另一种可能是使用视图并且组合视图列中的表列，以对此视图列表创建单个文本索引。这样可以避免进行两个不同的文本搜索调用。

组合文本列可提高性能。但是，这主要取决于单个搜索要求。

在外连接中使用文本搜索

如果执行一个使用 CONTAINS() 搜索函数的外连接查询，那么除非 CONTAINS() 谓词引用外连接的元组保留端上的表列，否则该查询可能会失败，并且会产生原因码 CTE0129 不允许将空值作为参数来传递。

例如，T1 是“T1 左外连接 T2”中的元组保留端，而 T2 是“T1 右外连接 T2”中的元组保留端。

搜索期间的性能注意事项

要在搜索期间提高性能，应考虑下列问题：

- 何时在 SQL 内进行搜索：
  - 如果注意到性能降低了，那么使用 explain 语句来检查 DB2 优化器的处理方案。
  - 参数搜索的搜索速度可以更快，尤其是当使用其他搜索谓词来减少结果大小时。
  - 如果不需要所有结果，那么使用结果限制关键字。
- 何时使用存储过程进行搜索：
  - 由于已将指定的高速缓存表表达式从数据库复制到内存中，因此，应确保工作站具有足够的内存可用于此数据。如果内存不足，那么会使用调页空间，这将降低搜索性能。
- 如果使用 NUMBEROFMATCHES 或 SCORE 函数而不使用 CONTAINS 函数，那么查询性能可能会下降。另外，为了避免重复处理，应确保 CONTAINS 函数中的字符串与 NUMBEROFMATCHES 或 SCORE 函数中使用的字符串完全匹配。

注： 有关最新的性能提示，请访问以下 DB2 Net Search Extender Web 站点：www.ibm.com/software/data/db2/extenders/netsearch/index.html

用户方案

在本章中可以通过使用下列逐步执行的示例来了解 Net Search Extender:

SQL 标量搜索示例
此命令行示例演示可用的建立索引和搜索功能。

存储过程示例
此命令行示例使用以上示例中的索引命令。添加了高速缓存后，该示例还演示可用于存储过程搜索的不同建立索引功能和搜索功能。

SQL 表值函数示例
SQL 表值函数示例是存储过程搜索示例的变体。

注: 在使用这些示例之前，通过使用安装验证过程来确保成功安装了 Net Search Extender。

SQL 标量搜索函数的简单示例

在 DB2 Net Search Extender 示例中使用下列步骤:

1. 创建数据库
2. 启用数据库以进行文本搜索
3. 创建表
4. 创建全文本索引
5. 装入样本数据
6. 使文本索引同步
7. 搜索文本索引

可以在操作系统的命令行上使用现有数据库发出样本命令。对于下列示例，数据库名称为 sample。

创建数据库
可以使用以下命令在 DB2 中创建数据库:

```
db2 create database sample
```

启用数据库以进行文本搜索
可以像在操作系统的命令行上发出 DB2 命令那样发出 DB2 Net Search Extender 命令。例如，使用以下命令来启动“Net Search Extender 实例服务”:

```
db2text START
```

然后准备数据库，使它与 DB2 Net Search Extender 配合使用:

```
db2text ENABLE DATABASE FOR TEXT CONNECT TO sample
```

对于每个数据库只需执行一次此步骤。

创建表

```
db2 "CREATE TABLE books (isbn VARCHAR(18) not null PRIMARY KEY,
     author VARCHAR(30), story CLOB(100k), year INTEGER)"
```

此 DB2 命令将创建一个称为 books 的表。它包含已出版的书籍的 author 列、story 列、isbn number 列和 year 列。注意，该表必须具有主键。

创建全文本索引

```
db2text "CREATE INDEX db2ext.myTextIndex FOR TEXT ON books (story)
                 CONNECT TO sample"
```

此命令将为 story 列创建全文本索引。该文本索引的名称为 db2ext.myTextIndex

装入样本数据

```
db2 "INSERT INTO books VALUES ('0-13-086755-1','John', 'A man was
          running down the street.',2001)"
db2 "INSERT INTO books VALUES ('0-13-086755-2','Mike', 'The cat hunts
          some mice.', 2000)"
db2 "INSERT INTO books VALUES ('0-13-086755-3','Peter', 'Some men
          were standing beside the table.',1999)"
```

这些命令将把这三本书的 isbn、author、story 和 publishing year 装入表中。

使文本索引同步

要使用 sample 表中的数据来更新文本索引，使用以下命令:

```
db2text "UPDATE INDEX db2ext.myTextIndex FOR TEXT CONNECT TO sample"
```

搜索文本索引

要搜索文本索引，使用以下 CONTAINS 标量搜索函数:

```
db2 "SELECT author, story FROM books WHERE CONTAINS
    (story, '\"cat\"') = 1 AND YEAR >= 2000"
```

注: 根据正在使用的操作系统 shell 的不同，在将文本搜索短语引起来的双引号前面可能需要添加不同的转义字符。以上示例将“\”用作转义字符。

此查询将搜索书籍的 year 值大于或等于 2000 并且包含单词 cat 的所有书籍。查询返回以下结果表:

```
AUTHOR Mike
STORY  The cat hunts some mice.
```

受支持的其他函数包括 SCORE 和 NUMBEROFMATCHES。SCORE 将返回一个指示符，指示搜索自变量描述找到的文档的仔细程度。NUMBEROFMATCHES 将返回在获得的文档中找到的查询词条匹配项的数量。

高速缓存用法和存储过程搜索的简单示例

在 DB2 Net Search Extender 存储过程搜索示例中使用下列步骤:

1. 使用高速缓存选项创建文本索引。
2. 使索引同步并激活高速缓存。
3. 使用 TEXTSEARCH 存储过程进行搜索。

注: 存储过程示例假定前一示例中的步骤是完整的，并且仍然启用了数据库。

使用高速缓存选项创建文本索引

由于已经启用了数据库，因此，使用以下命令来创建全文本索引:

```
db2text "CREATE INDEX db2ext.mySTPTextIndex FOR TEXT ON books (story)
                CACHE TABLE (author, story) MAXIMUM CACHE SIZE 1
                CONNECT TO sample"
```

在此示例中，全文本索引是针对 story 列的，并且它指定一个包含 author 和 story 列的高速缓存表。文本索引的名称为 mySTPTextIndex。

使索引同步并激活高速缓存

要使用插入到表中的数据来更新索引，使用以下命令:

```
db2text "UPDATE INDEX db2ext.mySTPTextIndex FOR TEXT CONNECT TO sample"
```

要激活高速缓存，使用以下命令:

```
db2text "ACTIVATE CACHE FOR INDEX db2ext.mySTPTextIndex FOR TEXT
                CONNECT TO sample"
```

这会将 author 和 story 列的内容装入到高速缓存中。

使用“TEXTSEARCH 存储过程”进行搜索

只有在某些情况下才能使用 DB2 Net Search Extender 存储过程。

```
 db2 "call db2ext.textSearch
               ('\"cat\"','DB2EXT','MYSTPTEXTINDEX',0,2,0,0,?,?)"
```

此查询将搜索有关 cat 的所有书籍，但是，将只返回前两个结果。书籍的结果表可能为如下所示:

```
Value of output parameters
--------------------------
Parameter Name  : SEARCHTERMCOUNTS
Parameter Value : 1
Parameter Name  : TOTALNUMBEROFRESULTS
Parameter Value : 1

AUTHOR    STORY
Mike      The cat hunts some mice.

Return Status = 0
```

有关搜索语法的更多样本，请查看 DB2 实例目录中的以下文件：sqllib/samples/extenders/db2ext/search

SQL 表值函数的简单示例

可以对在前面示例中创建的文本索引使用 SQL 表值函数。

SQL 表值函数查询对应于先前使用的 CONTAINS 查询。有关信息，请参阅第 67 页的『SQL 标量搜索函数的简单示例』主题下的“使文本索引同步”。

```
db2 "SELECT author, story FROM books b, table (db2ext.textsearch
          ('\"cat\"','DB2EXT','MYTEXTINDEX', 0, 2, CAST
          (NULL AS VARCHAR(18)))) T where T.primKey = b.isbn
```

在以上示例中，NULL 将被强制类型转换为主键的数据类型。

使用同义字来扩展搜索词条

不仅可以通过搜索特定搜索词条还可以搜索与其相关的词条来扩大查询范围。可以通过使用 Net Search Extender 的根据同义字查找并抽取相关搜索词条的功能来使此过程自动化。同义字是语义上相关的词条的受控词汇表，它通常涵盖特定主题区域。

Net Search Extender 允许您根据先前创建的同义字添加附加词条来扩展搜索词条。请参阅第 135 页的『搜索自变量的语法』以了解如何在查询中使用同义字扩展。

为了创建同义字以将其用于搜索应用程序，此时需要必须编译为内部格式的同义字定义文件，即同义字典。

本节描述以下内容:

- 『同义字的结构』

  同义字的结构类似根据关系链接在一起的节点网络。本节描述 Net Search Extender 的预定义关系以及如何定义您自已的关系。

- **第 71 页的『创建和编译同义字』**

  以下是对同义字定义文件语法的描述以及对用来将该文件编译为同义字典的工具的描述。

同义字的结构

同义字的结构类似根据关系链接在一起的节点网络。Net Search Extender 在同义字中查找词条的方法是从该词条开始，然后沿着通过该词条关系的路径发送在该过程中找到的词条。

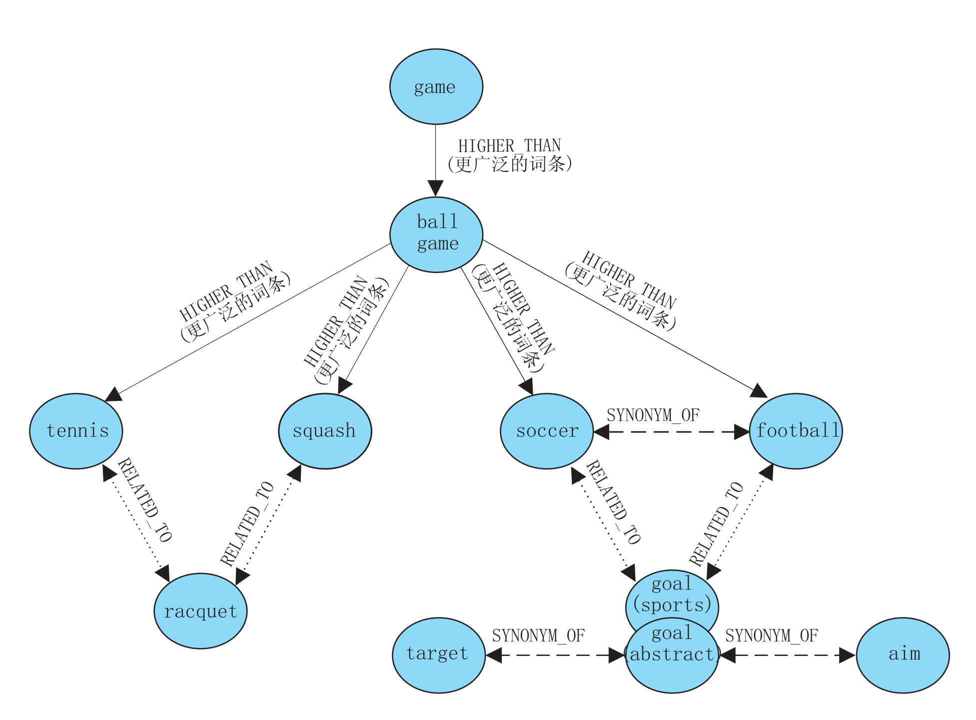


图 28. 同义字结构的示例

同义字条目是根据关系连接的。关系名（例如，BROADER）允许您将扩展限制为关系层次结构中的某些命名行。某些关系是双向的，其他是非双向的；例如，BROADER 是非双向关系的名称。

预定义的同义字关系

以下是 Net Search Extender 中预定义的关系:

- **关联关系**

  关联关系是两个词条之间的双向关系，这两个词条并不表示同一概念，但是彼此相关。

预定义关联关系：RELATED_TO

示例：

```
tennis RELATED_TO racket
football RELATED_TO goal (sports)
```

- **同义词关系**

  同义词关系是两个词条之间的双向关系，这两个词条有相同或相似的含义并且可以彼此替代。例如，此关系可以用在词条与其缩写之间。

  预定义同义词关系：SYNONYM_OF

  示例：

  ```
  spot SYNONYM_OF stain
  US   SYNONYM_OF United States
  ```

  第 70 页的图 28 中的图显示相同同义字中的两个 goal 词条。一个使用注释 (sports) 指定，另一个使用注释 (abstract) 指定。即使词条拼写相同，同义词关系仍可连接不同词组。可以通过在定义同义字时使用不同的关系来形成这一点。
- **层次结构关系**

  层次结构关系是在两个词条之间的非双向关系，一个词条比另一个词条具有更广泛（更普遍）的意义。根据关系的方向，可以使用该关系来查找更专业或更普遍的词条。

  预定义层次结构关系：
  - LOWER_THAN 用来形成缩小关系

    LOWER_THAN 关系用来形成更专业的词条序列。随着搜索关系的范围越来越小，词条变得越具体。例如，如果沿着 LOWER_THAN 关系查找词条 ball game，那么结果在越来越专业的词条的列表中可能包括 squash tennis 等等。
  - HIGHER_THAN 用来形成扩大关系。

    HIGHER_THAN 关系用来形成更普遍的词条序列。随着这种关系的搜索范围越来越大，词条将变得越不具体。例如，如果沿着 HIGHER_THAN 关系查找词条 ball game，那么结果在越来越普遍的词条列表中可能包括 game 等等。

定义您自己的关系

Net Search Extender 允许您定义自己的 RELATED_TO、LOWER_THAN 和 HIGHER_THAN 同义字关系。因为每个关系名必须是唯一的，所以必须通过添加唯一编号来限定这种关系名，如：RELATED_TO(42)。

可以使用相同关系编号来定义不同类型的关系，例如 LOWER_THAN(42)。编号 0 用来表示 Net Search Extender 预定义的关系。

创建和编译同义字

执行下列步骤来创建可以由 Net Search Extender 功能使用的同义字：

1. 创建同义字定义文件。
2. 将定义文件编译为同义字典。

创建同义字定义文件

要创建您自己的同义字，第一步是使用文本编辑器在定义文件中定义它的内容。

限制。文件名（包括扩展名）的长度不能超过 256 个字符。同一目录中可以有几个同义字，但是建议每个同义字具有不同的目录。

提供了样本英语同义字定义文件 `nsesamplethes.def`。Windows 系统的同义字目录为:

```
<sqllib>\db2ext\thes
```

在 UNIX 系统上，同义字目录为:

```
<instance_owner_home>/sqllib/db2ext/thes
```

以下是该文件中的前几个定义组:

```
:WORDS
      accounting
  .RELATED_TO account checking
  .RELATED_TO sale management
  .SYNONYM_OF account
  .SYNONYM_OF accountant

:WORDS
      acoustics
  .RELATED_TO signal processing

:WORDS
      aeronautical equipment
  .SYNONYM_OF turbocharger
  .SYNONYM_OF undercarriage

:WORDS
      advertising
  .RELATED_TO sale promotion
  .SYNONYM_OF advertisement
:
:
:
```

图 29. 样本同义字定义文件的片断

有关每个定义组的语法，请参阅第 73 页的『同义字支持』。

必须将每个成员写至单个行。必须在每个相关联的词条前面添加关系名。如果成员词条彼此相关，那么指定成员关系。

成员词条和相关联词条的长度限制为 64 个字符。同一字母的单字节字符和双字节字符被认为是相同的。不区分大写和小写字母。词条可以包含空白字符，并且可以使用单字节字符句点“.”或冒号“:”。

用户定义的关系都是以*关联*类型为基础的。它们由 1 与 128 之间的唯一数字标识。

将定义文件编译为同义字典

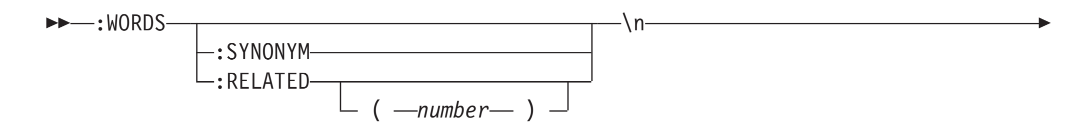
要编译同义字定义文件，运行 `db2extth` 命令。

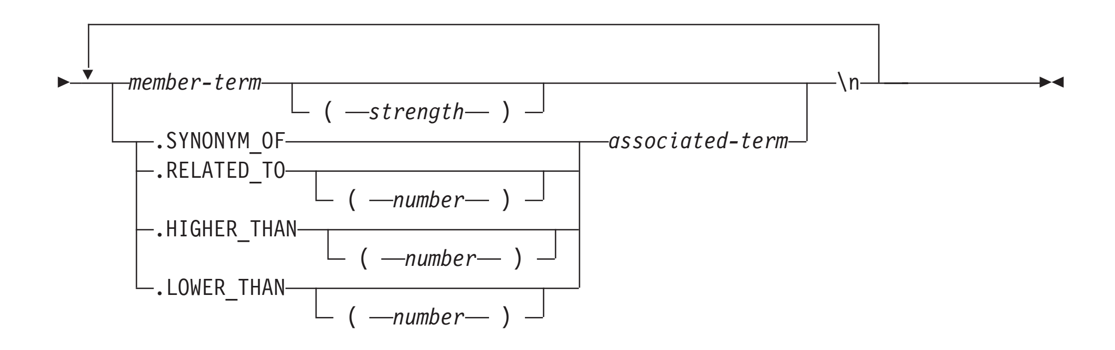
要在分区环境中使用同义字典，确保所有物理节点都可访问创建的文件。

同义字支持

以下是当您创建自己的同义字时每个定义组的语法:

同义字定义的语法





注意，\n 不是语法的一部分，而是表示同义字定义文件中一行的结束。

可以在同义字定义文件中插入注释行，如下所示:

```
# my comment text
```

:WORDS

开始一组相关单词的关键字。

:SYNONYM, :RELATED [(*number*)],

一个关系名。

关系名由关系类型和数字组成。如果省略了数字，那么假定它为零，这是系统提供的关系名。:SYNONYM 始终是系统提供的关系名。

以冒号开头的关系名（例如，:SYNONYM）在彼此相关的一系列单词前面添加同一关系。例如:

```
:WORDS
  :SYNONYM
      air steward
      cabin staff member
      flight attendant
```

member-term

要包括在同义字典中的一个词条。

- 最大长度为 64 个字节（对于代码页 UTF-8，最大长度为 42 个字节）。
- 同一字母的单字节字符和双字节字符被认为是相同的。
- 不区分大小写字符。
- 词条可以包含空格字符。
- 不能使用单字节字符句点“.”或冒号“:”。

如果想要同义字查找不包括与查找的词条关系不大的单词，那么此参数就非常有用。强度是 1 到 100 之间的数值。缺省值为 100。

.SYNONYM_OF, .RELATED_TO [(*number*)], .HIGHER_THAN [(*number*)], .LOWER_THAN [(*number*)]

一个关系名。关系名 .HIGHER_THAN 对应于 BROADER 查询关系，而 .LOWER_THAN 对应于 NARROWER 查询关系。关系名由关系类型和数字组成。如果省略了数字，那么假定它为零，这是系统提供的关系名。关系名 .SYNONYM 始终是系统提供的关系名。

以句点开头的关系名（例如，.SYNONYM_OF）定义一个单词与另一个单词之间的关系。例如:

```
:WORDS
      air steward
  .SYNONYM_OF cabin staff member
  .SYNONYM_OF flight attendant
```

可选的 *number* 标识用户定义的关系。这必须是整个同义字定义文件中的唯一数字（当前为 1 到 128）。例如: `RELATED_TO(42)`。

如果想要在应用程序中使用同义字关系的符号名称而不是关系名和数字，那么应用程序必须处理名称至数字的映射。例如，如果将关系 `opposite_of` 定义为 RELATED_TO(1)，那么应用程序必须将此名称映射至内部关系名 RELATED_TO (1)。

associated-term

必须在每个相关联的词条前面添加关系名。相关联的词条根据指定的关系来与每个成员词条相关。如果所有成员词条都互相相关，那么可以使用成员关系来指定此参数。

- 最大长度为 64 个字节（对于代码页 UTF-8，最大长度为 42 个字节）。
- 同一字母的单字节字符和双字节字符被认为是相同的。
- 不区分大小写字符。
- 词条可以包含空格字符。
- 不能使用单字节字符句点“.”或冒号“:”。

以下是相关联词条的一个示例:

```
:WORDS:SYNONYM
    reject
    decline
        RELATED_TO(1) accept
```

同义字支持的 CCSID

同义字支持下列 CCSID:

819 Latin 1

850 PC 数据 Latin 1

874 泰国语

932 组合的日语

943 组合的日语

949 组合的韩国语

950 组合的繁体中文

954 日语

970 组合的韩国语

1208 UTF 8

1250 Latin 2

1252 Latin 1

1253 捷克语

1254 土耳其语

1255 希伯莱语

1256 阿拉伯语

1258 越南语

1363 组合的韩国语

1381 组合的简体中文

1383 中文（简体），组合的 SBCS/DBCS

1386 中文（简体），组合的 SBCS/DBCS

5039 日语（组合的 SBCS/DBCS）

同义字工具返回的消息

ADM_MSG_INVALID_CCSID

指定了无效的 CCSID。

请求的代码页不受支持。

ITL_THES_MSG_BUFFER_OVERFLOW

缓冲区溢出。

ITL_THES_MSG_DICT_EXIST

同义字典 *字典名* 已存在。

不能覆盖。

ITL_THES_MSG_DICT_INTEGRITY_ERROR

字典 *字典名* 不完整。

同义字典文件已毁坏。

ITL_THES_MSG_DICT_NOT_EXIST

同义字典 *字典名* 不存在。

ITL_THES_MSG_DICT_VERSION_ERROR

字典 *字典名* 版本错误。

同义字典是使用不兼容的早期版本创建的。

ITL_THES_MSG_ERROR_IN_FILE

文件 *文件名* 中存在错误。

ITL_THES_MSG_FILE_ACCESS_ERROR

未能访问文件 *文件名*。

ITL_THES_MSG_FILE_CLOSE_ERROR

未能关闭文件 *文件名*。

ITL_THES_MSG_FILE_EOF_ERROR

文件名 中出现文件意外结束。

定义文件中存在错误。

ITL_THES_MSG_FILE_OPEN_ERROR

打不开文件 *文件名*。

ITL_THES_MSG_FILE_REACHED_END

同义字定义文件 中出现文件意外结束。

定义文件中存在错误。

ITL_THES_MSG_FILE_READ_ERROR

未能读取文件 *文件名*。

ITL_THES_MSG_FILE_REMOVE_ERROR

未能除去文件 *文件名*。

ITL_THES_MSG_FILE_RENAME_ERROR

未能将文件 *文件名 1* 重命名为 *文件名 2*。

ITL_THES_MSG_FILE_WRITE_ERROR

未能写文件 *文件名*。

ITL_THES_MSG_IE_BLOCK_START

在文件 *文件名* 的第 *行号* 行中找不到任何块起始行。

ITL_THES_MSG_IE_EMPTY

同义字定义文件 *文件名* 是空的。

ITL_THES_MSG_IE_NO_TERM

在 *文件名* 的第 *行号* 行中未定义任何词条。

ITL_THES_MSG_IE_REL_SYNTAX

在 *文件名* 的第 *行号* 行中不正确地指定了关系。

ITL_THES_MSG_IE_STRENGTH_DOMAIN

强度超出范围。

有效值为 1 - 100; 缺省值为 100。

ITL_THES_MSG_IE_STRENGTH_SYNTAX

不正确地指定了强度值。

语法: 在该词条后输入 [:20] 表示强度为 20。

ITL_THES_MSG_IE_TERM_LEN

同义字词条长度超过 64 个字符。

ITL_THES_MSG_IE_USER_DEF

在 *文件名* 的第 *行号* 行中不正确地指定了关系。

ITL_THES_MSG_IE_USER_DEF_DOMAIN

关系编号超出 *文件名* 第 *行号* 行的范围。

ITL_THES_MSG_INPUT_ERROR

同义字定义文件 *文件名* 的第 *行号* 行存在错误。

ITL_THES_MSG_INTERNAL_ERROR

内部错误。

ITL_THES_MSG_LOCKED

正在使用同义字典 *字典名*。

ITL_THES_MSG_LOCKING_ERROR

未能锁定字典 *文件名*。

ITL_THES_MSG_MEMORY_ERROR

内存错误。

ITL_THES_MSG_NAMELEN_ERROR

参数错误 *文件名*。同义字定义文件名过长。

ITL_THES_MSG_NO_TARGET_DIR_ERROR

参数错误。未指定任何目标目录。

ITL_THES_MSG_NONAME_ERROR

参数错误。未指定任何同义字定义文件名。

ITL_THES_MSG_NORMALIZE_ERROR

规范化词条时出错。

同义字定义文件中存在错误。

ITL_THES_MSG_OUTFILE_EXIST

输出文件 *文件名* 已存在。

ITL_THES_MSG_PARAMETER_ERROR

内部参数错误。

ITL_THES_MSG_PATHLEN_ERROR

参数错误 *文件名*。同义字定义文件路径过长。路径长度不能超出操作系统中的目录名支持的最大长度。

ITL_THES_MSG_UNEXPECTED_ERROR

出现内部意外错误。

文本搜索引擎

Net Search Extender 提供以下文本搜索引擎信息:

- 标记化
- 停用词
- 配置

标记化

在建立索引期间，Net Search Extender 用以下方法处理文档文本：将文本分成标记。

单词

使用所有字母数字字符（“a”至“z”、“A”至“Z”和“0”至“9”）来创建全文本索引。分隔符是空白字符和下面的句子识别部分中描述的字符。对控制字符（例如，换行符（又称为换行字符）和空白字符）进行如下解释：行中间的控制字符（小于 0x20）被视为空白字符。忽略换行（0x0A）之前和之后的空白字符和控制字符。将 1 字节字符前后的换行视作是空白字符，而始终将表示同一字符的 2 字节字符视作相同字符。如果在搜索期间未指定任何内容，那么将同一字符的大写字母和小写字母（例如，“A”和“a”）视作相同字符；如果在搜索期间需要精确匹配，那么将它们视作不同字符。

语句

Net Search Extender 将后跟空白字符的“.”、“!”和“?”以及行末尾处的日语和中文句点识别为句子结束。

段

段识别与文档格式有关。在纯文本格式中，任何两个连续的换行符（可能带有插入回车符）被识别为段边界。在 HTML 中，段标记 <p> 解释为段边界。其他文档格式不支持段识别。

停用词

停用词是在文本检索过程中出现频率很高但又没有相关内容的单词。通常所有功能词（从语言学角度来看）都被认为是停用词，例如，“and”、“or”和“in”。搜索停用词的索引将严重降低文本检索系统的精度。

Net Search Extender 为多种语言提供了停用词处理。可以在创建索引时设置配置参数 IndexStopWords，该配置参数可确定是否对停用词建立了索引。缺省值为 1，表示对停用词建立了索引。

如果不想对停用词建立索引，那么必须将 IndexStopWords 设置为 0，并在创建索引时使用语言参数来指定用于输入文档的语言。如果未对停用词建立索引，那么索引将更小并且速度更快。在创建索引之后，不要在配置 `.ini` 文件模板中改变此值，这样做会导致根据为文档创建索引的时间来以不同方式处理文档，并因此而导致不连贯地处理停用词。

仅当文档集合中的所有文档都使用同一种语言时，才能在建立索引期间忽略停用词。

支持停用词的语言

下列语言提供停用词处理。

AR_AA
: 在阿拉伯国家或地区使用的阿拉伯语

CA_ES
: 在西班牙使用的加泰罗尼亚语

DA_DK
在丹麦使用的丹麦语

DE_CH
在瑞士使用的德语

DE_DE
在德国使用的德语

EL_GR
在希腊使用的希腊语

EN_GB
在英国使用的英语

EN_US
在美国使用的英语

ES_ES
在西班牙使用的西班牙语

FI_FI 在芬兰使用的芬兰语

FR_CA
在加拿大使用的法语

FR_FR
在法国使用的法语

HE_IL 在以色列使用的希伯莱语

IS_IS 在冰岛使用的冰岛语

IT_IT 在意大利使用的意大利语

IW_IL 在以色列使用的希伯莱语

NB_NO
在挪威使用的挪威博克马尔语

NL_BE
在比利时使用的荷兰语

NN_NO
在挪威使用的挪威尼诺斯克语

PT_BR
在巴西使用的葡萄牙语

PT_PT
在葡萄牙使用的葡萄牙语

RU_RU
在俄罗斯使用的俄语

SV_SE
在瑞典使用的瑞典语

配置

Net Search Extender 能够搜索其字符用在不同组合中的单词，例如，字母数字、数字和特殊字符。为此，Net Search Extender 提供了下列配置:

字符规范化

字符规范化确保能够搜索到那些可以两种方式书写的单词。例如，德语单词“Überbau”还可以书写为“Ueberbau”。规范化确保使用“Überbau”或“Ueberbau”都可搜索这两个单词。此功能还会将重音字母（例如，“accès”）规范化为相匹配的单个字符（例如，“acces”）。

使用特定字符作为单词的一部分

通过使用特定字符作为单词的一部分，可以确保可能涉及到一系列字母数字字符、特殊字符和数字的产品名可以作为单个单词来搜索。例如，通过将字母数字组合“DT9”看作一个单词，或者通过启用“/”特殊字符，就可以将 AS/400® 或 OS/390® 作为一个完整单词来搜索，而不是作为“AS”和“400”或者“OS”和“390”来进行搜索。

对于这些配置设置提供了开关。要定制开关，在创建索引之前更改 `.ini` 文件模板。

`.ini` 文件模板存储在 `sqllib/db2ext/cteixcfg.ini` 中。因为还可以使用 `CREATE INDEX` 命令来更改此模板文件中的大多数值，所以建议您只更改下列值:

AccentRemoval（用于字符规范化）
UmlautNormalization（用于字符规范化）
TreatNumberAsWords（将数字字符看作单词的一部分）
AdditionalAlphanumCharacters（将特定字符用作单词的一部分）

AccentRemoval

此参数指定是否将重音字符规范化为相匹配的单个字符。例如，还可以将 événement 的索引创建为 evenement。缺省值为 true。

UmlautNormalization

此参数指定是否会将一个元音变音字符的索引创建为具有相同含义的两个字符。例如，还会将“Übersee”的索引创建为“Uebersee”。缺省值为 true。

TreatNumbersAsWords

此参数指定一个单词旁边的数字字符是否是该单词的一部分。例如，将“DT9”当作一个单词，而不是当作一个单词“DT”和一个数字“9”。

AdditionalAlphanumCharacters

此参数的字符串值定义将哪些字符当作一个单词的一部分。一串特殊字符必须是采用 UTF-8 编码的一个或多个字符组成的序列。缺省字符串包含字符“/-@”。

在用作部分单词的字符列表中不允许使用通配符 % 和 _。如果使用，将在执行查询时导致产生问题。

如果想更改任何这些配置值，请在创建索引之前编辑 `.ini` 文件。要激活不活动的开关，请除去那一行开头的“;”。有关进一步的信息，请参阅 `cteixcfg.ini` 文件。

建议您不要改变 `.ini` 文件中的任何其他值。

第 8 章 使用结构化文档

Net Search Extender 使您可以对结构化文档中的文本或数字字段（例如，标题、作者或价格）建立索引并进行搜索。文档可以是 XML、Outside In 或 HTML 格式，也可以包含用户定义的标记（GPP）。

在*文档模型*中使用标记（markup tag）及其字段名来定义为文档中的哪些字段建立了索引，从而使这些字段可供搜索。可以在对字段的查询中使用该字段的名称（也称为节名）。

为了能够在这些字段中进行搜索，当创建包含文档的文本索引时，必须指定 `FORMAT` 和 `MODEL` 文件。

搜索本机存储的 XML 文档

下列各节将描述对存储在本地的 XML 文档进行搜索。将告诉您如何将小节搜索的概念应用于存储在本地的 XML 文档，以及如何将此功能集成到 XQuery 处理中。

通常，对 XML 数据列创建索引时，不需要指定格式（FORMAT）。缺省情况下，当对 XML 列创建文本索引时，Net Search Extender 将选择 XML 格式。不允许对 XML 数据列使用格式说明符 TEXT 和 HTML。

在用来说明对 XML 列创建和使用文本索引的后续样本中，使用了以下 XML 文档。该 XML 文档存储在 t1 表的 c2 列（类型为 XML）中。

```
<?xml version="1.0">
<purchaseOrder orderDate="2001-01-20">
   <shipAddress countryCode="US">
      <name>Alice Smith</name>
        <street>123 Maple Street</street>
        <city>Mill Hill</city>
        <zip>90999</zip>
   </shipAddress>
   <item partNo="123" quantity="1">
        <name>S&B Lawnmower Type ABC-x</name>
        <price>239.90</price>
      <shipDate>2001-01-25</shipdate>
   </item>
   <item partNo="987" quantity="1">
        <name>Multifunction Rake ZYX</name>
        <price>69.90</price>
      <shipDate>2001-01-24</shipdate>
   </item>
</purchaseOrder>
```

使用缺省文档模型

如果在 CREATE INDEX 语句中没有指定任何文档模型，那么 Net Search Extender 将使用缺省文档模型。

缺省文档模型的一个特征就是：节名在用于指定每个元素和属性的绝对路径的 XPath 表示法中。注意，搜索查询中的节名并不是在执行查询期间求值的 XPath 表达式。它们是用来表示结构化文档中的特定部分（元素和属性）的一些名称。

如果您未在使用模型文件，那么按如下所示为 XML 文档定义文本索引:

```
db2text CREATE INDEX i1 FOR TEXT ON t1(c2) CONNECT TO mydbname
```

由于 c2 列的数据类型为 XML，因此可以省略 FORMAT 规范。在此情况下，FORMAT 规范缺省设置为 XML。

当未指定文档模型时，对于每个 XML 元素，都将根据它在文档中的绝对 XPath 来自动为它指定一个名称。例如，在搜索查询中，通过节名 /purchaseOrder/item/price 就可以访问 price 元素。使用节名 /purchaseOrder/shipAddress/@countryCode 就可以访问 countryCode 属性。

在使用 db2text update 命令更新索引之后，将 SECTION 搜索与标量搜索函数配合使用的 SQL 表达式可能如下所示:

```
SELECT c2 FROM t1
WHERE CONTAINS(c2, SECTIONS("/purchaseOrder/item/name") "Rake") = 1
```

查询将返回如上所示的样本 XML 文档。

使用定制文档模型

如果想定义定制的节名，那么必须指定一个模型文件来对文档的某些部分指定用户定义的名称。使用文档模型的一个优点是：可以指定您想要对 XML 文档的哪些部分建立索引，并使用 XPath 表达式来指定这些部分。

以上 XML 文档的模型文件可能如下所示:

```
<?xml version="1.0"?>
<XMLModel>
   <XMLFieldDefinition
      name="itemName"
      locator="/purchaseOrder/item/name" />
   <XMLFieldDefinition
name="customerName"
      locator="//shipAddress/name" />
   <XMLAttributeDefinition
name="partNumber"
type="NUMBER"
      locator="/purchaseOrder//item/partNo" />
   <XMLFieldDefinition
name="none"
      locator="/purchaseOrder/orderDate"
      exclude="yes" />
</XMLModel>
```

注意，文档模型为上面的搜索查询中所引用的 /purchaseOrder/item/name 元素指定名称 itemName。

使用模型文件的索引定义为:

```
CREATE INDEX i1 FOR TEXT ON t1(c2) DOCUMENTMODEL XMLModel IN
   /mydir/myfilename/xmlmodel.xml CONNECT TO mydbname
```

（使用 DOCUMENTMODEL 参数指定的）文档模型名指定模型文件中的根元素。这是 XML 文档模型的 XMLModel。路径 /mydir/ ... 指向用于定义模型的文件。

文档模型语法支持一部分 W3C XPath 语法，而 W3C XPath 语法可以很方便地标识元素。

在使用上述模型文件创建文本索引并且使用 db2text update 命令更新索引之后，就可以按如下所示搜索 /purchaseOrder/item/name 元素:

```
SELECT c2 FROM t1
WHERE CONTAINS(c2, SECTIONS("itemName") "Rake") = 1
```

注意与搜索查询的差别，搜索查询未指定文档模型。但是这两种查询都会返回上面所提到的同一样本 XML 文档。

XML 文档模型还会对 item 元素的 XML 属性 partNo 定义一种 partNumber 属性。Net Search Extender 属性定义的数据类型必须始终是 NUMBER。

上述样本模型文件中的属性定义允许对如下所示的值范围进行搜索:

```
SELECT c2 FROM t1  WHERE CONTAINS
  (c2, ATTRIBUTE "partNumber" BETWEEN 300 AND 500) = 1
```

XQuery 支持

当在数据库中搜索 XML 文档时，还可以使用 XQuery 来处理搜索结果。通过利用 DB2 的混合数据库引擎，可以将 SQL 文本搜索查询与 XQuery 处理组合在一起。

这是通过在 XQuery 上下文中使用 db2-fn:sqlquery() 输入函数来完成的。要使用 XQuery 输入函数，必须使用 set language XQuery 命令从 SQL 切换到 XQuery，或者在查询前面添加关键字 XQuery 作为前缀。对于解析器来说这是一个重要的指示符，它与 XQuery 表达式一起工作，并且必须遵循应用于 XQuery 语言的区分大小写规则和语法规则。

db2-fn:sqlquery() 函数采用用来表示全查询的字符串文字。db2-fn:sqlquery() 函数将返回一个 XML 序列，该序列表示由全查询选择的 XML 列值的并置结果。

可以使用以下表达式来对存储在本地的 XML 文档同时进行文本搜索和 XQuery 处理:

```
XQUERY db2-fn:sqlquery('SELECT c2 FROM t1
                    WHERE CONTAINS(c2,
           ''SECTIONS ("/purchaseOrder/item/name") "Rake" '')
     = 1 ')//shipAddress/name
```

上述查询将返回包含名为“Rake”的采购订单项的 XML 文档中 shipAddress 元素下的所有 name 元素。必须在 SELECT 语句中显式选择 XML 列（在上述示例中为 c2）。

可以按如下所示通过将 FLWOR 构造嵌入到应用程序中来扩展上述样本:

```
XQUERY FOR $item in db2-fn:sqlquery('SELECT c2 FROM t1
           WHERE CONTAINS(c2, '' SECTIONS ("/purchaseOrder/item/name") "Rake" '')
       = 1 ')
           WHERE $item[@partNo > "800"]
RETURN $item/price
```

注意，对输入函数 db2-fn:sqlquery() 进行全查询时始终都会返回完整的 XML 文档，该 XML 文档中包含产生的匹配项。

考虑存储在本地数据库中的以下 XML 文档:

```
<?xml version="1.0"?>
<dept bldg="101">
  <employee id="901">
     <name>Sabine</name>
     <resume>DB2 programmer</resume>
```

```
  </employee>
  <employee id="902">
     <name>Holger</name>
     <resume>XML expert</resume>
  </employee>
</dept>
```

用于搜索您所在的部门中其履历包含“XML”词条的职员的语句可能如下所示:

```
SELECT c2 FROM t1  WHERE CONTAINS(c2, SECTIONS("/dept/employee/resume") "XML")=1
```

上述搜索查询将返回完整的 XML 文档。按如下所示将该搜索查询嵌入到 XQuery 中:

```
XQUERY db2-fn:sqlquery('SELECT c2 FROM t1
                 WHERE CONTAINS(c2,
        ''SECTIONS ("/dept/employee/resume") "XML" '') =1') //employee/name
```

它将返回以下两个结果:

```
<name>Sabine</name>
<name>Holger</name>
```

注意，尽管职员 Sabine 的履历中没有词条“XML”，但是她还是出现在上述 XQuery 中的结果序列中。之所以如此，是因为该全查询将返回整个文档，即，它将返回完整的 XML 文档，而该 XML 文档中至少有一个职员的履历中具有词条“XML”。

如果想查询只返回结果 `<name>Holger</name>`，那么发出以下 XQuery 语句:

```
XQUERY for $d in db2-fn:sqlquery('SELECT c2 FROM t1
                 WHERE CONTAINS(c2,
        ''SECTIONS ("/dept/employee/resume") "XML" '') =1')
   return §d/dept/employee/name[contains(parent::employee/resume,"XML")];
```

通过对 XML 列使用对结构敏感的全文本索引，Net Search Extender 就可以滤出在 `/dept/employee/resume` 节中具有词条“XML”的所有 XML 文档。根据所返回的 XML 文档子集，并且通过使用 XPath 轴来浏览 XML 文档，return 语句 `return §d/dept/employee/name[contains(parent::employee/resume,"XML")]` 将只返回满足以下条件的那些 <name> 元素：在它们的名为 <resume> 的同代元素中具有 XML。

结构化文档支持

文档模型如何描述结构化文档

采用 HTML 或 XML 格式的文档是结构化文档的示例，它们包含用来标识文本字段和文档属性的标记。文本字段可以包含一些类似于文档的标题、作者或描述之类的信息。

以下是从结构化纯文本文档中抽取的一部分。它包含由类似于 HTML 的标记定界的文本。

```
[head]Handling structured documents
[/head]

[abstract]This document describes the concept of structured documents
and the use of document models to...
[/abstract]:
:
```

当 Net Search Extender 为结构化文档创建索引时，它必须识别结构才能为文本字段和属性创建索引，并将它们与唯一名称存储在一起。这使得 Net Search Extender 通过使用 SECTION 或 ATTRIBUTE 子句可以有选择地搜索特定文本字段，或者查找具有特定属性的文档。

为了使 Net Search Extender 能够了解特定文档格式的结构，必须将*文档模型*中结构的定义传递给 Net Search Extender。也可以使用 Net Search Extender 提供的缺省文档模型。

当调用 `CREATE INDEX` 命令来为文档创建索引时，将文档模型的名称指定为自变量。例如，`CREATE INDEX i1 FOR TEXT ON t1(c2) DOCUMENT MODEL GPPModel IN mymodel.xld CONNECT TO db`

参数 `GPPModel` 表示您正在使用的文档模型的类型。

在您可以使用文档模型来为文档创建索引之前，必须首先定义文档模型，然后让索引知道该文档模型。

注: 如果 XML 文档使用了格式错误的索引，那么建立索引的过程将在文档中遇到错误的地方停止。这表示将仅对部分文档建立索引。如果不更正文档，那么将只能在建立索引的那部分文档中进行搜索。仅当表列类型不为XML 时才会发生这种情况。

文档模型的示例

必须为您打算创建索引的每种文档格式定义一种文档模型。以下是纯文本结构化文档的简单文档模型。注意，该示例中的 GPP 代表“通用解析器”。

```
<?xml version="1.0"?>
<GPPModel>                  - the GPP document model begin here

    <GPPFieldDefinition       - a field definition begins here
    name="Head"               - the name you assign to this field
      start="[head]"           - the boundary string at the beginning of the field
    end="[/head]"             - the boundary string at the end of the field
  exclude="YES" />

    <GPPFieldDefinition       - the next field definition begins here
    name="Abstract"
    start="[abstract]"
    end="[/abstract]"
    exclude="NO" />
:
:
</GPPModel>
```

文档模型在 XML 语言中是使用第 96 页的『文档模型参考』中定义的标记指定的。文档模型由文本字段定义和属性定义组成。以上示例说明在 `GPPFieldDefinition` 元素中只定义了文本字段定义。类似地，可以使用 `GPPAttributeDefinition` 来定义文档属性。

该示例中的第一行 `<?xml version="1.0"?>` 指定文档模型是使用 XML 标记来编写的。每个文本字段定义都指定边界字符串以标识源文档中的字段定义的 `start` 和 `end`。因此，每当文档包含后跟一些文本的字符序列 `[head]`，然后接着是字符序列 `[/head]`，就会将这些边界字符串之间的文本作为由名称 `head` 标识的文本字段的内容。

对每个字段定义指定字段名。查询可通过此字段名并借助 CONTAINS 函数中的 SECTION 子句来将搜索范围限制为某个文本字段的内容。该字段的名称可以是固定的，也可以按照规则从结果单元的内容来派生。例如，这样一个名称可以是 XML 实体的标记名或者是 XML 属性的名称。

文档模型

文档模型主要控制需要对文档结构的哪些部分建立索引以及如何对它们建立索引。其目的为:

- 标识应在源文档中区别出来的文本字段
- 确定这样的文本字段的类型
- 向文本字段指定字段名

当文档模型将文本标识为属于文本字段时，将认为该文本是文档的文本内容的一部分，并将抽取一些词条并将它们存储在索引中。

文档模型的元素随用于该文档格式的解析器的不同而不同:

- 对于 HTML 格式，文档模型使用 HTML 标记名来定义应对哪些标记建立索引以及如何处理元标记信息。
- 对于 XML 格式，没有预定义的标记集，所以文档模型必须首先定义哪些标记是有用的。对于名称相同的 XML 元素，还可以根据它们嵌入哪些其他元素中来区分。
- 对于 GPP（通用解析器）格式，文档模型与解析器进行更进一步的交互，原因是它必须确定文本字段的边界。此处字段定义必须指定用于检测字段边界的字符串。
- 对于 Outside In 格式，文档模型使用类似 HTML 标记名的标记来定义应对哪些标记建立索引以及如何处理元标记信息。注意，“Outside In 变换技术”又称为 INSO。

文本字段

文档模型允许您将文档部分或节标识为属于特定文本字段和/或标识为文档属性。

对文档的文本全部建立索引，而不管它是否为文本字段的一部分。将有意义的词条抽取并存储在索引中。这意味着非限制文本搜索包括对该文本的搜索。

但是，通过定义文本字段，可以在特定字段中有选择性地搜索文本。例如，可以在文本字段 `Abstract` 中搜索包含单词 `structure` 的文档。例如，`SELECT doc from my_docs WHERE CONTAINS (doc, SECTIONS(Abstract) "structure" = 1`。

文本字段可以在文档中多次出现。例如，可以定义一个包含所有图形标题的文本字段。一个文本字段还可以被另一个文本字段覆盖。

如果想避免为某些文本字段的内容创建索引，可以指定包含 `exclude="YES"` 的字段定义。可以在第 98 页的『文本字段和文档属性的局限性』中查找文本字段和文档属性的限制的列表。

文档属性

文档属性包含类型为 `number` 的简短格式化信息。与文本字段相比，可以使用值范围来搜索包含这些属性的文档。

属性不是与已建立索引的文本存储在一起的，而是存储在一个单独的项索引中。因此，要按属性的内容来搜索文档，必须对属性显式地进行属性搜索。例如，`SELECT doc FROM my_docs WHERE CONTAINS (doc,ATTRIBUTE "year" BETWEEN 2001 AND 2005) = 1`。

数字属性:

Net Search Extender 提供了一个识别浮点数的解析器。下面是属性值的正确和不正确格式的一些示例。

表 6. 受支持的属性值格式

正确格式	不正确格式
1000 1 000 1.000 – 句点是一个十进制 字符	1,000
100 000 100 000.00123	1 000 000 – 1 和 0 之间存在两个空格

注意，在一个数字的十进制小数中不允许存在空格字符。例如，1 000.000 100 将被当作两个数 1000.000 和 100 来对待。

不支持特定于语言的分隔符和特定于语言的货币格式。

缺省文档模型

对于 HTML、XML 和 Outside In 的已过滤文档，Net Search Extender 提供在创建索引期间没有指定文档模型的情况下使用的缺省文档模型。对于结构化纯文本文档，必须提供并指定文档模型。

如果使用其中一种缺省文档模型，那么:

- 将对所有字段建立索引，并且不会抽取任何特殊信息（如元信息）。
  - 对于 HTML 和 INSO 格式，将为每个字段指定相应标记的名称。
  - 对于 XML，XML 文档的所有 XML 节点将映射至重叠字段，这些重叠字段是由相应节点的标准元素路径标识的。例如，路径 `/play/role/name`。
- 不会对数字属性建立索引（原因是未在缺省文档模型中定义任何数字属性）。

表 7. 受支持文档格式的缺省文档模型的行为

文档类型	缺省文档模型的行为
HTML	接受下列标记作为文本字段：<a> <address> <au> <author> <h1> <h2> <h3> <h4> <h5> <h6> <title>。字段名是标记名，例如“address”。
XML	接受所有标记作为文本字段。字段名是标准元素路径名，例如，“/play/title”。
结构化纯文本（GPP）	没有缺省文档模型。
Outside In（INSO）	接受由 Outside In 过滤器返回的第 94 页的『定义 Outside In 过滤文档的文档模型』中显示的文档属性作为文本字段。字段名是 Outside In 使用的文档属性的名称，例如：“SCCCA_TITLE”。

对于每种类型的文档，都定义了缺省文档模型。由于每种模型都是不相同的，因此，下列各节中为每种模型都提供了示例和解释。

注:

尽管缺省文档模型的确正确地处理文档，但为了更好地建立索引和搜索，您应该定义自已的文档模型。

借助缺省文档模型，对文档的文本全部建立索引，而不管它是否为文本字段的一部分。这意味着非限制文本搜索包括对该文本的搜索。

定义纯文本结构化文档的文档模型

以下是文档模型元素的参数:

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。使用上面的示例，您可在名为 `Abstract` 的文本字段中搜索包含单词 `structure` 的文档。

start 代码页 UTF-8 中的边界字符串，它标记文本字段或文档属性的开始。指定边界字符串没有任何规则；它们可以是任意 UTF-8 字符串。以下是一些示例: start=″introduction:″、start=″note!″ 和 start=″$$...″。

必须使用缺省 XML 字符条目（“<”表示“<”，而“&”表示“&”）来指定不可打印字符和特殊 XML 字符“<”和“&”。

end 可选。代码页 UTF-8 中的边界字符串，它标记文本字段或文档属性的结束。如果不指定结束标记，那么假定找到的下一个开始标记作为该字段的结束。如果找不到任何后继开始标记，那么该字段将一直扩展至文档结尾，并且不会再标识任何其他字段。

type 文档属性的类型必须始终是“NUMBER”。该参数不适用于字段定义。

exclude
YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

在该示例中，将排除字段定义“head”，但是将包括定义“abstract”。

限制:

- 不能有两个字段定义或属性定义的开始标记相同。但是，字段定义和属性定义可以有相同的开始标记和结束标记。
- 一个开始标记不能恰好是另一个开始标记的前缀。例如，不能有一个开始标记是“author”而另一个开始标记是“authority”的情况。
- 开始标记和结束标记不能是空字符串。

对 GPP 文档建立索引时发生的情况

通用解析器将扫描文档以查找其中一个开始边界字符串。当它找到一个开始边界字符串时，它将对后续字段进行语法分析，直到找到相应的结束边界字符串为止。

然后将根据定义词条对字段的内容建立索引，即，作为文本字段或文档属性。如果文本字段和文档属性具有相同的开始和结束边界字符串，那么将把字段的内容同时作为文本字段和文档属性来对其建立索引。

不允许字段的嵌套；如果在到达结束边界字符串前发现新的开始边界字符串，那么会将新的开始边界字符串解释为普通文本。

如果未找到相应的结束边界字符串，那么假定该字段扩展至文档结束，并将报告适当的原因码。

如果在文档模型中未指定结束边界字符串，那么新的开始边界字符串指示先前字段的结束。

定义 HTML 文档的文档模型

HTML 解析器将文本转换为代码页 UTF-8。它执行 HTML 标记识别，并将它们归类为下列标记类:

- 要忽略的已标记信息，如字体信息
- 提供位置信息的标记，如表示新段的 <p>;
- 提供结构信息的标记，如 <Title>

它将识别 HTML 4 中定义的所有字符实体引用（如“ä”（ä））并将它们解析为 UTF-8 中相应的代码点。

它将识别元标记并对元标记文本进行语法分析。

以下是 HTML 文档的示例:

```
<HTML>
<HEAD>
<META NAME="year" CONTENT="2002">
<TITLE> The Firm </TITLE>
</HEAD>
<BODY>
<H1>Synopsis</H1>;


<H1>Prologue</H1>;:
:
</BODY>
```

以下是 HTML 文档模型的示例:

```
<?xml version="1.0"?>
<HTMLModel>

  <HTMLFieldDefinition
  name="subtitle"
  tag="title"
   exclude="NO" />

  <HTMLFieldDefinition                - This is the start of text field
  name="header1"
  tag="h1"
  exclude="YES" />                    - This is the end of the text field

  <HTMLAttributeDefinition            - This is the start of the document
  name="year"                                  attribute
  tag="meta"
  meta-qualifier="year"
  type="NUMBER" />                    - This is the end of the document
                 attribute
  </HTMLModel>
```

第一行 `<?xml version="1.0"?>` 指定使用 XML 标记编写文档模型。注意，此模型不是为 XML 格式文档编写的。

每个字段都是在 HTMLFieldDefinition 或 HTMLAttributeDefinition 标记内定义的，该标记包含元素参数。

所有文本字段定义都必须包含在 <HTMLModel> 标记内。在创建索引期间，标记名称是作为一个参数来传递的：CREATE INDEX iA FOR TEXT ON T1(C2) DOCUMENTMODEL HTMLModel IN myModel.xml CONNECT TO db。

以下是文档模型元素的参数:

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。使用上面的示例，您可在名为 subtitle 的文本字段中搜索包含单词 firm 的文档。

tag 标识其开始和（暗示的）结束标记标识文本字段或文档属性的元素。该名称的元素内的文本组成已定义字段的内容。

忽略标记的大小写。

使用上面的示例，将任何 H1 标记之后的文本作为字段“header1”的一部分建立索引。根据样本文档，将对“synopsis”和“prologue”建立索引。

meta-qualifier

此标记必须与 **tag** 元素配合使用。通过指定 tag=″meta″，将抽取与 meta-qualifier 相匹配的内容的值。

在 HTML 文档示例中，元标记具有下列元素:

```
<META NAME="year" CONTENT="2002">
```

在文档模型示例中，那么为 meta-qualifier=″year″。因此，内容“2002”将作为属性“year”的值来建立索引。

type 文档属性的类型必须是“NUMBER”。该参数不适用于字段定义。

exclude

YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

在该示例中，将排除字段定义“header1”，但是将包括定义“subtitle”。

将对文档的所有其他文本建立索引，但不作为任何字段的一部分。

元素参数

以下是文档模型元素的参数:

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。使用上面的示例，您可在名为 subtitle 的文本字段中搜索包含单词 firm 的文档。

tag 标识其开始和（暗示的）结束标记标识文本字段或文档属性的元素。该名称的元素内的文本组成已定义字段的内容。

忽略标记的大小写。

使用上面的示例，将任何 H1 标记之后的文本作为字段“header1”的一部分建立索引。根据样本文档，将对“synopsis”和“prologue”建立索引。

meta-qualifier

此标记必须与 **tag** 元素配合使用。通过指定 tag="meta"，将抽取与 meta-qualifier 相匹配的内容的值。

在 HTML 文档示例中，元标记具有下列元素：

```
<META NAME="year" CONTENT="2002">
```

在文档模型示例中，那么为 meta-qualifier="year"。因此，内容“2002”将作为属性“year”的值来建立索引。

type 文档属性的类型必须是“NUMBER”。该参数不适用于字段定义。

exclude

YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

在该示例中，将排除字段定义“header1”，但是将包括定义“subtitle”。

将对文档的所有其他文本建立索引，但不作为任何字段的一部分。

定义 XML 文档的文档模型

XML 文档的文档模型允许您定义如何将在 XML 文档中找到的元素映射至字段和/或文档属性。

以下是 XML 文档的示例：

```
<?xml version="1.0"?>
<purchaseOrder orderDate="2001-01-20">             [4]
        <shipAddress countryCode="US">                [1]
                <name>Alice Smith</name>                  [2]
                <street>123 Maple Street</street>
                <city>Mill Hill</city>
                <state>CA</state>
                <zip>90999</zip>
        </shipAddress>
        <item partNo="123" quantity="1">           [3]
                <name>S&B Lawnmower Type ABC-x</name>
                <price>239.90</price>
                <shipDate>2001-01-25</shipDate>
        </item>
        <item partNo="987" quantity="1">     [3]
                <name>Multifunction Rake ZYX</name>
                <price>69.90</price>
                <shipDate>2001-01-24</shipDate>
        </item>
</purchaseOrder>
```

以下是与上面的样本文档相匹配的 XML 文档模型的示例：

```
<?xml version="1.0"?>
<XMLModel>

<XMLFieldDefinition                   [1]
name="addresses"
locator="/purchaseOrder/shipAddress" />

<XMLFieldDefinition                  [2]
name="customerName"
locator="//shipAddress/name"
exclude="yes" />

<XMLAttributeDefinition                  [3]
name="partNumber"
```

```
type="NUMBER"
locator="/purchaseOrder//item/@partNo" />

<XMLFieldDefinition                  [4]
name="none"
locator="/purchaseOrder/@orderDate" />

</XMLModel>
```

第一行 <?xml version="1.0"?> 指定使用 XML 编写模型。每个字段都是在 XMLFieldDefinition 或 XMLAttributeDefinition 标记内定义的，它包含元素参数。

注意，所有文本字段定义都必须包含在 <XMLModel> 标记内。在创建索引期间，此标记名称是作为一个参数来传递的：CREATE INDEX i1 FOR TEXT ON T1(C2) DOCUMENTMODEL XMLModel in myModel.xml CONNECT TO db。

使用与示例模型文件中的定义相对应的数字标记了样本中的字段和属性。

允许嵌套字段，例如，如果一个规范的 XPath 位置选择位于由另一个属性定义选择的 XML 元素中的节点。嵌套字段显示在上面的样本 XML 文档中。addresses 字段将选择 XML 文档中的一个节点，该节点将控制由 customerName 字段选择的节点。因此，该嵌入节点的内容在逻辑上同时属于这两个字段。尽管文本字段可以是重叠的，但是，只能对这些字段中的文本建立一次索引。在此示例中，当在带有字段限制进行搜索时，将同时在 addresses 和 customerName 中找到 Alice Smith。但是，由于定位器表达式的相匹配语义，不能将同一个 XML 节点映射至多个字段。

Net Search Extender 不会尝试检测 XML 文档的代码页。将采用 DB2 代码页。

字段的内容由下列规则确定：

- 对于其定位器与注释、处理指令或 XML 属性相匹配的字段，字段内容为实际注释文本、处理指令文本或者属性值文本。
- 对于与 XML元素或根节点相匹配的字段，字段内容由来自任何嵌入元素（与具有规范 exclude="YES" 的字段相匹配的元素除外）的任何文本组成。

文档必须包含格式良好的 XML，但对于要在 XML 文档中指定的 DTD，则不必如此。不进行 DTD 验证或外部实体解析；Net Search Extender 只针对文档模型来匹配 XML 文档。内部实体在需要时被 XML 替代。

元素参数

以下是文档模型元素的参数：

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。

可在名称中使用下列变量的其中一个。该变量将由源文档中的匹配元素生成的字符串来替换。

Variable
值

$(NAME)
与 XPath 路径相匹配的 XML 元素的实际限定名（QName）。

$(LOCALNAME)
与 XPath 相匹配的 XML 元素的实际局部名（不带前缀）。

$(PATH)
实际绝对路径，以与 XPath 相匹配的 XML 元素的斜杠和标记序列组成（用作缺省文档模型中的名称）。

type 文档属性的类型必须是"NUMBER"。该参数不适用于字段定义。

locator

XPath 语言中的表达式，它们选择要用作搜索字段的源文档的各个部分。

当编写"XML 文档模型"文件时，`locator` 内的限定名（称为 QNames）必须与 XML 文档中的某些标记完全相同，否则，将不能识别任何字段，并且对字段的查询将不会返回结果。

下面是一些定位器的示例。

`purchaseOrder | salesOrder`
所有 `purchaseOrder` 元素和 `salesOrder` 元素

`shipAddress`
所有 `shipAddress` 元素

***** 所有元素（它是 `child::*` 的缩写 – 有关进一步的信息，请参阅语法）

name/item
具有 `name` 父代的所有 `item` 元素

purchaseOrder//item
具有 `purchaseOrder` 祖代的所有 `item` 元素

/ 根节点

comment()
所有注释节点

processing-instruction()
所有处理指令

attribute::*（或者 @*）
所有属性节点

文字为括在单引号或双引号内的字符串。有关终端标记的精确定义，请参阅 XML 建议。

Net Search Extender 文档模型支持的 XPath 定位器与"XML 样式表语言变换"（XSLT）模式相似。它们正是由不包含任何谓词或函数"id"和"key"或节点测试"text()"和"node()"的 XSLT 模式的子集组成。

ignore

YES 或 NO。使用该参数来建立定位器的异常。

有时您可能想要指定常规定位器（例如 *）以与想要对其建立索引的节点相匹配。但是您还可指定不应对与较特定定位器相匹配的某些节点建立索引。

为此，为在建立索引期间要忽略的节点的较特定定位器加上字段定义。于是就对此定位器指定了比常规定位器更高的优先级（请参阅下面的内容），并指定 `ignore="yes"`。这将指示索引器一定不要为匹配节点生成字段信息。

注意，当这种被忽略节点嵌入在字段生成节点中时，将对被忽略节点的内容建立索引，原因是它也属于字段生成节点的内容。

priority

-1 和 +1 之间的浮点数，它指定要给予由特定定位器找到的定义的优先级。

如果不指定优先级，将使用缺省优先级：

- 将由 | 分隔的多个备用项视作一组定义，每个备用项一个定义。
- 与单个名称相匹配的定位器；即，下列任一格式的定位器具有缺省优先级 0:
  – ChildOrAttributeAxisSpecifier QName
  – ChildOrAttributeAxisSpecifier processing-instruction(Literal))
- 格式为 ChildOrAttributeAxisSpecifier NCName:* 的定位器具有缺省优先级 -0.25。
- 格式为 ChildOrAttributeAxisSpecifier NodeTest 的其他定位器具有缺省优先级 -0.5。
- 任何其他定位器具有缺省优先级 0.5。

注意定位器越具体，缺省优先级越高。例如，非特定定位器 * 给予找到的定义低优先级，而名称较特定的定位器给予较高的优先级。

另外，注意当一个节点与多个定位器相匹配时，您可以通过对它们指定优先级来确定选择哪个定义。将选择具有最高优先级的定义。如果两个定义具有相同的优先级，将选择最新的定义。

此冲突解决与在"XML 样式表语言变换"（XSLT）中使用的完全相同。

exclude

YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

在该示例中，将排除字段定义"customerName"，但是将包括定义"addresses"。

定义 Outside In 过滤文档的文档模型

Outside In 格式的文档模型允许您将由给定标记集标识的结构化元素映射至 Net Search Extender 文本字段和文档属性，在这一点上它们与 HTML 文档模型很相似。假定有一组 Microsoft Word 文档并且您想要将文档属性"title"、"subject"和"keyword"作为字段建立索引，并且将文档属性"author"和"category"作为文档属性建立索引。Outside In 文档模型的以下示例将完成此映射：

```
<?xml version="1.0"?>
<INSOModel>

<INSOFieldDefinition
name="title"
tag="SCCCA_TITLE"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_SUBJECT"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_KEYWORDS"/>

<INSOAttributeDefinition
name="author"
```

```
tag="SCCCA_AUTHOR"
type="STRING"/>

<INSOAttributeDefinition
name="category"
tag="SCCCA_CATEGORY"
type="STRING"/>

</INSOModel>
```

元素参数

以下是文档模型元素的参数:

name 对文本字段或文档属性指定的名称。对每个字段定义指定字段名，并对每个属性定义指定属性名。查询可通过使用这些名称将搜索限制为某个文本字段的内容并可搜索具有某个属性的文档。

tag 标识一个标记，其开始和结束或暗示结束元素标记文本字段或文档属性。该名称的元素内的文本组成已定义字段或属性的内容。忽略标记的大小写。下面描述了可能的值。

type 文档属性的类型可以是"NUMBER"、"DATE"或"STRING"。此参数不适用于字段定义。

exclude
YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

Outside In 文档模型由字段和/或属性定义组成，它们每一个都定义一个名称和一个标记。对于属性定义，还需要一个类型，而字段定义具有可选"exclude"标志。对于 HTML 模型，这种定义的名称属性定义将向其映射文档部分的 Net Search Extender 字段或属性的名称。它可以是任意 UTF-8 文本字符串。要获取更多信息，请参阅 Outside In Content Access Specification V7.5。

有关与 Outside In 开始、结束和文档属性标记相关的标记属性的可能值的列表，请参阅以获取进一步的信息。

对 Outside In 文档建立索引时发生的情况

缺省情况下，将所有文本作为不属于任何字段的内容建立索引。每当出现在文本流中的开始标记与当前活动的文档模型中的定义词条匹配时，就将根据该定义词条来处理在开始标记与其相应结束标记之间的文本。例如，作为已索引字段和已排除字段和/或属性来处理。

如果不存在匹配定义，那么将忽略开始标记及其相应的结束标记。

Outside In 过滤器自动识别文档的格式和代码页，CCSID 规范不起任何作用。如果 Outside In 过滤器无法确定正确的格式和代码页，那么会将文档视作 ASCII 文件。

文档模型参考

Net Search Extender 提供了有关文档模型的以下参考信息:

- 文档模型的 DTD
- 定位器（XPath）表达式的语义
- 文本字段和文档属性的限制
- Outside In 标记属性值

文档模型的 DTD

以下是文档类型定义（DTD）形式的文档模型语法的正式描述:

```
<!ELEMENT GPPModel (GPPFieldDefinition|GPPAttributeDefinition)+>
<!ELEMENT HTMLModel (HTMLFieldDefinition|HTMLAttributeDefinition)+>
<!ELEMENT XMLModel (XMLFieldDefinition|XMLAttributeDefinition)+>

<!ELEMENT GPPFieldDefinition EMPTY>
<!ATTLIST GPPFieldDefinition name CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition start CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition end CDATA #IMPLIED>
<!ATTLIST GPPFieldDefinition exclude (YES|NO) NO>

<!ELEMENT GPPAttributeDefinition EMPTY>
<!ATTLIST GPPAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition start CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition end CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT HTMLFieldDefinition EMPTY>
<!ATTLIST HTMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLFieldDefinition exclude (YES|NO) NO>

<!ELEMENT HTMLAttributeDefinition EMPTY>
<!ATTLIST HTMLAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT XMLFieldDefinition EMPTY>
<!ATTLIST XMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition locator CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition ignore (YES|NO) NO>
<!ATTLIST XMLFieldDefinition priority CDATA #IMPLIED>
<!ATTLIST XMLFieldDefinition exclude (YES|NO) NO>

<!ELEMENT XMLAttributeDefinition EMPTY>
<!ATTLIST XMLAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST XMLAttributeDefinition locator CDATA #REQUIRED>
<!ATTLIST XMLAttributeDefinition ignore (YES|NO) NO>
<!ATTLIST XMLAttributeDefinition priority CDATA #IMPLIED>
<!ATTLIST XMLAttributeDefinition type NUMBER #REQUIRED>
```

定位器（XPath）表达式的语义

根据 XML 数据模型，XML 文档被视作包含下列类型节点的树:

- 根节点
- 元素节点

- 文本节点
- 属性节点
- 名称空间节点
- 处理指令节点
- 注释节点

这些节点之间的链接（换句话说就是树形成关系）反映 XML 文档中的直接包含关系。

根节点只能出现在根处，而不能出现在树的其他位置。它包含作为其子代的文档元素和可选注释及处理指令。

元素节点可包含任何类型的节点，但根节点除外。其他类型的节点只允许作为树的叶节点。

有三种类型的**包含链接**：“子代”、“属性”和“名称空间”。“属性”和“名称空间”包含链接必须分别引导至属性和名称空间节点。换言之，要访问元素节点的子代（按照图论），需要跟随“属性”链接来查找所有包含的属性，跟随“名称空间”链接来查找所有包含的名称空间声明，跟随“子代”链接来查找包含的元素、文本节点、处理指令和注释。

需要根据上下文节点来解释 XPath 表达式，并表示一组节点。当用作 Net Search Extender 选择器模式时，上下文节点空闲，也就是说，相对路径模式 `p` 解释为 `//p`。

以下是一些 Net Search Extender XPath 选择器模式：

- 上下文 N 中的 `Pattern '|' LocationPathPattern` 表示 Pattern 和 LocationPathPattern（都在上下文 N 中）与之相匹配的节点的并集。
- 上下文 N 中的 `'/'RelativePathPattern` 表示此 RelativePathPattern 在根上下文中表示的任何内容。
- 上下文 N 中的 `'//'RelativePathPattern` 表示在作为根的子代（在子代轴上）的任何上下文中解释的这一 RelativePathPattern 表示法的并集。
- 当且仅当上下文 N 中的节点与其父代的上下文中的 StepPattern 相匹配，且其父节点与上下文 N 中的 RelativePathPattern 相匹配时，`RelativePathPattern '/' StepPattern` 才与该节点相匹配。
- 当且仅当上下文 N 中的节点与其父代的上下文中的 StepPattern 相匹配，且它的祖代节点与上下文 N 中的 RelativePathPattern 相匹配时，`RelativePathPattern '//' StepPattern` 才与该节点相匹配。
- 上下文 N 中的 `'child'::NodeTest`（缩写语法：NodeTest）与作为 N 的子代（在子代轴上）且符合 NodeTest 的节点相匹配。
- 上下文 N 中的 `'attribute'::NodeTest`（缩写语法：@NodeTest）与作为 N 的属性且符合 NodeTest 的节点相匹配。
- 当且仅当节点为指定类型时，`NodeType '(' ')'` 才适合该节点。
- `'processing-instruction' '(' Literal ')'` 适合将 Literal 作为其名称的任何处理指令类型节点。
- `'*'` 适合任何元素或属性节点（元素名称的名称屏蔽）。
- `NCName ':' '*'` 适合将 NCName 作为其名称前缀的任何元素节点。
- `QName` 适合具有指定名称的任何节点。

注意

格式为 NameTest 的 NodeTest 假定节点为所选轴上的主体类型，它是属性轴上的属性类型，也是子代轴上的子类型。因此，NameTest 不能用来选择注释或处理指令节点，而只能选择子节点和属性节点。而且，该模式允许选择任何类型的节点（名称空间节点除外），原因是不允许使用 Axis 说明符“namespace”。

模式示例:

- `chapter | appendix` 表示所有章节元素和附录元素
- `table` 表示所有表元素
- `*` 表示所有元素（注意，这是子元素 ::* 的缩写）
- `ulist/item` 表示具有 ulist 父代的所有词条元素
- `appendix//subsection` 表示具有附录祖代的所有子节元素
- `/` 表示只包含根节点的单元素集合
- `comment()` 表示所有注释节点
- `processing-instruction()` 表示所有处理指令
- `attribute::*`（或 `@*`）表示所有属性节点

以下是定位器元素的语法:

```
Locator     ::= LocationPathPattern
                      | Locator '|'  LocationPathPattern
  LocationPathPattern  ::= '/' RelativePathPattern ?
           | '//'? RelativePathPattern
  RelativePathPattern  ::= StepPattern
                      | RelativePathPattern '/' StepPattern
                      | RelativePathPattern '//' StepPattern
  StepPattern    ::= ChildOrAttributeAxisSpecifier NodeTest
  ChildOrAttributeAxisSpecifier ::=
                      ('child' | 'attribute') '::'
                      | '@'?
  NodeTest    ::= NameTest
                      | NodeType '(' ')'
                      | 'processing-instruction' '(' Literal ')'
  NameTest    ::= '*' | NCName ':' '*' | QName
  NodeType    ::= 'comment' | 'processing-instruction'
```

NCName 和 QName 是按“XML 名称建议”定义的:

NCName
: 不包含冒号的 XML 名称

QName
: 前面可以加上 NCName 后跟冒号的 NCName。例如: `NCName:NCName`

文本字段和文档属性的局限性

下面列示的是文本字段和文档属性存在的局限性:

- 索引中的字段的最大数目: 32767
- 一个文档中类型为 STRING 的一个属性的值的最大数目: 1024
- 类型为 STRING 的属性的最大数目: 253
- STRING 属性值中的字符数截断为 128
- 类型为 DATE 和 NUMBER 的属性的最大数目: 32766

- DATE 或 NUMBER 属性值中的字符数截断为 128
- 对于 NUMBER 属性，接受双精度浮点数作为值。
- 可对一个文档中类型为 DATE 或 NUMBER 的一个属性指定的值的最大数目：无限制

下列各项是可包括在 HTML 文档模型中的标记:

- <A>
- <ADDRESS>
- <AU>
- <AUTHOR>
- <H1>
- <H2>、<H3>、<H4> 和 <H5>
- <H6>
- <TITLE>

不能在 HTML 文档模型中将可包含其他标记的标记（如 <HEAD> 和 <BODY>）指定为文本字段。

Outside In 标记属性值

与 Outside In 文档属性标记类型相关的标记属性的可能值:

```
SCCCA_ABSTRACT
SCCCA_ACCOUNT
SCCCA_ADDRESS
SCCCA_ATTACHMENTS
SCCCA_AUTHORIZATION
SCCCA_BACKUPDATE
SCCCA_BASEFILELOCATION
SCCCA_BILLTO
SCCCA_BLINDCOPY
SCCCA_CARBONCOPY
SCCCA_CATEGORY
SCCCA_CHECKEDBY
SCCCA_CLIENT
SCCCA_COMPANY
SCCCA_COMPLETEDDATE
SCCCA_COUNTCHARS
SCCCA_COUNTPAGES
SCCCA_COUNTWORDS
SCCCA_CREATIONDATE
SCCCA_DEPARTMENT
SCCCA_DESTINATION
SCCCA_DISPOSITION
SCCCA_DIVISION
SCCCA_DOCCOMMENT
SCCCA_DOCTYPE
SCCCA_EDITMINUTES
SCCCA_EDITOR
SCCCA_FORWARDTO
SCCCA_GROUP
SCCCA_KEYWORD
SCCCA_LANGUAGE
SCCCA_LASTPRINTDATE
SCCCA_LASTSAVEDBY
SCCCA_MAILSTOP
SCCCA_MANAGERSCCCA_MATTER
SCCCA_OFFICE
```

```
SCCCA_OPERATOR
SCCCA_OWNER
SCCCA_PRIMARYAUTHOR
SCCCA_PROJECT
SCCCA_PUBLISHER
SCCCA_PURPOSE
SCCCA_RECEIVEDFROM
SCCCA_RECORDEDBY
SCCCA_RECORDEDDATE
SCCCA_REFERENCE
SCCCA_REVISIONDATE
SCCCA_REVISIONNOTES
SCCCA_REVISIONNUMBER
SCCCA_SECONDARYAUTHOR
SCCCA_SECTION
SCCCA_SECURITY
SCCCA_SOURCE
SCCCA_STATUS
SCCCA_SUBJECT
SCCCA_TITLE
SCCCA_TYPIST
SCCCA_USERDEFINEDPROP
SCCCA_VERSIONDATE
SCCCA_VERSIONNOTES
SCCCA_VERSIONNUMBER
```

与 Outside In 开始和结束标记子类型相关的标记属性的可能值:

```
SCCCA_ALTFONTDATA
SCCCA_ANNOTATIONREFERENCE
SCCCA_CAPTIONTEXT
SCCCA_CHARACTER
SCCCA_COMPILEDFIELD
SCCCA_COUNTERFORMAT
SCCCA_CUSTOMDATAFORMAT
SCCCA_DATEDEFINITION
SCCCA_DOCUMENTPROPERTYNAME
SCCCA_ENDNOTEREFERENCE
SCCCA_FONTANDGLYPHDATA
SCCCA_FOOTNOTEREFERENCE
SCCCA_FRAME
SCCCA_GENERATEDFIELD
SCCCA_GENERATOR
SCCCA_HYPERLINK
SCCCA_INDEX
SCCCA_INDEXENTRY
SCCCA_INLINEDATAFORMAT
SCCCA_LISTENTRY
SCCCA_MERGEENTRY
SCCCA_NAMEDCELLRANGE
SCCCA_REFERENCEDTEXT
SCCCA_STYLE
SCCCA_SUBDOCTEXT
SCCCA_TOA
SCCCA_TOAENTRY
SCCCA_TOC
SCCCA_TOCENTRY
SCCCA_TOF
SCCCA_VECTORSAVETAG
SCCCA_XREF
```

注意，这些表包括任何文档属性以及 INSO 过滤器识别的所有标记子类型。有两个子类型例外：SCCCA_DOCUMENTPROPERTY 和 SCCCA_BOOKMARK。

第 9 章 参考

实例所有者的管理命令

本节描述实例所有者的管理命令的语法。实例所有者管理包括检查 Net Search Extender 锁定和更新服务的状态，以及启动和停止这些服务。

这些命令是 DB2TEXT 命令的子命令，并且允许管理特定于 DB2 实例的 Net Search Extender 服务。

命令	用途
『CONTROL 命令』	列示和删除全文本索引锁定。还将列示高速缓存状态。
第 102 页的『START 命令』	启动 Net Search Extender 实例服务。
第 103 页的『STOP 命令』	停止 Net Search Extender 实例服务。

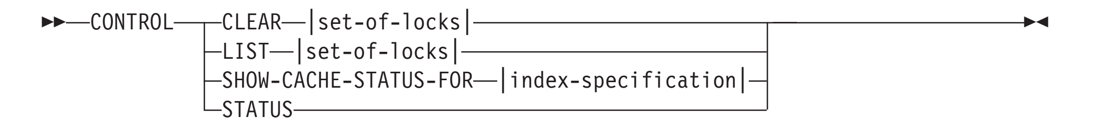
CONTROL 命令

用途

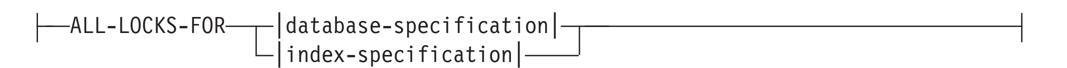
此命令允许您列示和删除由“Net Search Extender 实例服务”管理的全文本索引锁定。如果锁定和更新服务正在运行，那么可以查看它们的状态以及有关已激活高速缓存的信息。

在分布式 DB2 环境中，这仅影响当前分区。用户负责为期望的分区调用 DB2 命令 db2_all。

命令语法



set-of-locks:



index-specification:



Database-specification:

```
├──DATABASE──database-name──────────────────────────────────────────┤
```

命令参数

CLEAR
: 使用 CLEAR 来强制清除一组锁定。在检查导致锁定问题的原因后谨慎地使用此命令。

  如果有任何索引管理命令（例如，更新索引）仍然处于活动状态，那么不要使用 CLEAR 命令。

LIST
: 使用 LIST 来获取关于对特定索引或数据库持有的当前锁定的信息。如果有更新锁定，那么该命令还会提供有关迄今为止已处理的文档数的信息。

  注意，这只适用于锁定持有索引的时候。

  当使用复制捕获表时，不会执行任何更新操作。但是，对为其创建索引的源表执行的插入或更新操作可能会引起插入操作。

set-of-locks
: 仅在指定的数据库或索引中使用锁定。

SHOW CACHE STATUS FOR
: 显示指定索引的高速缓存表的激活状态。可以是"Not Activated"或"Currently Activated"。如果已激活高速缓存，那么它将显示关于高速缓存使用情况的详细信息。例如，最大高速缓存大小（以兆字节计）、要插入的最大文档数和保留在高速缓存表中的空间（以千字节计）。

STATUS
: 通过使用 STATUS 关键字，该命令显示锁定和更新"Net Search Extender 实例服务"是否已启动并正在运行。

DATABASE *database-name*
: 正在使用的服务器上的数据库的名称。

INDEX *index-schema*.*index-name*
: 目前正在使用的文本索引的模式和名称。这是在 CREATE INDEX 命令中指定的。

用法说明

当管理命令错误消息指示存在锁定问题时，确保没有任何冲突任务正在运行。例如，在 `UPDATE` 命令运行时尝试运行 ALTER 命令。

对增量索引更新使用 SHOW CACHE STATUS FOR，以检查指定内存大小是否仍然大到足以保存下一次更新期间的所有更新信息，或检查是否已经完成激活。

START 命令

用途

此命令将启动守护程序，它控制 DB2 服务器上的全文本索引的锁定以及全文本索引的自动更新。

注: 由于该命令不会对索引激活任何临时高速缓存表，所以在可以使用存储过程开始搜索之前需要执行单个 ACTIVATE CACHE 命令。

权限

必须作为 DB2 实例所有者在服务器或分布式 DB2 环境中的任何一个服务器上运行此命令。

命令语法

```
▶▶─START─────────────────────────────────────────────────────────▶◀
```

命令参数

无

用法说明

在 Windows 上，该命令将启动 db2ext-<InstanceName> 服务。还可以使用常规 window 方法启动此命令。但是，无法通过终端服务客户机启动 Net Search Extender。

如果 START 失败，那么调度程序中仍可能有引用不再存在的索引的过时条目。编辑 ../sqllib/db2ext/ctedem.dat 文件并除去任何过时条目。重新运行 START 命令。

在成功启动 Net Search Extender 之后，进程 ctelock（Windows 上的 ctelock.exe）在系统上就处于活动状态。而在 UNIX 机器上，会创建若干共享资源（共享内存和信号量）并将它们存储在 /tmp 目录中。Net Search Extender 需要这些文件，并且只要实例在运行，就不能删除这些文件。但是，如果在执行 STOP 之后未成功完成 START 命令，那么务必除去 /tmp 目录中所有旧的锚点文件。所有用户都必须能够读、写和执行 /tmp 目录。

要锁定全文本索引，可以修改配置文件以满足您的要求。

STOP 命令

用途

此命令将停止 Net Search Extender 的锁定和更新服务。

权限

必须作为 DB2 实例所有者在服务器或分布式 DB2 环境中的任何一个服务器上运行此命令。

命令语法

```
▶▶─STOP─┬───────┬────────────────────────────────────────────────▶◀
        └─FORCE─┘
```

命令参数

FORCE

即使进程现在持有锁定或对任何索引激活了高速缓存表，仍然会停止服务。如果不指定 FORCE，那么该命令在这些情况下将会失败，并且会提供有关活动高速缓存的警告。

用法说明

停止 Net Search Extender 实例服务之后，就不会允许您继续使用特定的 Net Search Extender 命令。如果先前将已激活的高速缓存与索引配合使用，那么在重新启动服务时，必须再次激活临时高速缓存。

在成功停止 Net Search Extender 之后，进程 ctelock（Windows 上的 ctelock.exe）就已终止。还会删除 UNIX 机器上 /tmp 目录中的共享资源和锚点文件。

数据库管理员的管理命令

本节描述数据库管理员的管理命令的语法。数据库管理包括设置数据库以供 Net Search Extender 使用然后禁用此设置。

只有 ENABLE DATABASE 和 DISABLE DATABASE 命令才是 DB2TEXT 命令的变体，即使所有这些命令都允许在数据库级别进行管理。

命令	用途
『ENABLE DATABASE 命令』	启用当前数据库以创建全文本索引。
第 105 页的『DISABLE DATABASE 命令』	复位 Net Search Extender 为数据库完成的准备工作。
第 107 页的『DB2EXTHL 命令』	缺省 UDF 使用 100 KB 的文档并返回 200 KB 的 CLOB。

提示

如果没有将任何数据库连接信息指定为 db2text 命令的一部分，那么 db2text 可执行文件将会建立与在环境变量 DB2DBDFT 中指定的缺省子系统的隐式连接。

ENABLE DATABASE 命令

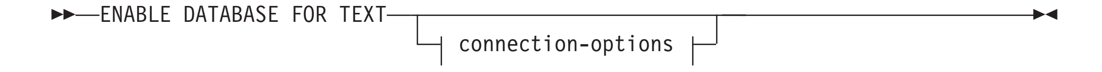
用途

此命令使数据库能够对文本列创建和利用全文本索引。

权限

必须作为数据库管理员运行此命令才能启用数据库。这要求您具有 SYSADM 权限以便能够将 DBADM 授予 DB2 实例所有者。

命令语法



connection-options:

```
├──┬─────────────────────────────────────────────────────────┬──────┤
   └─CONNECT TO─database-name─┬────────────────────────────┬─┘
                              └─USER─userid─USING─password─┘
```

命令参数

CONNECT TO *database-name*
: 作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 环境变量并且用户使用具有必需的 DB2 权限的用户标识运行该命令，那么可以省略此参数。

USER *userid* **USING** *password*
: 使用 *userid* 和 *password* 来连接至数据库。

用法说明

此命令准备已连接的数据库供 Net Search Extender 使用。经过此必要步骤，才可对数据库中的表/列创建 Net Search Extender 索引。

可以通过使用 DB2EXT.DBDEFAULTS 目录视图来查看运行该命令后建立的数据库缺省值。

对数据库的更改

此命令将 DBADM 权限授予与已启用数据库的 DB2 实例关联的 DB2 实例所有者。

ENABLE DATABASE 命令在 DB2EXT 模式中创建各种数据库对象，如 Net Search Extender 目录、UDF 和存储过程。在运行该命令后，下列目录视图可用:

```
db2ext.dbdefaults
db2ext.textindexes
db2ext.textindexformats
db2ext.indexconfiguration
```

还要注意，上面的表位于称为 IBMDEFAULTGROUP 的数据库的缺省表空间中。它分布在 db2nodes.cfg 中定义的所有节点上。

对文件系统的更改

无。

DISABLE DATABASE 命令

用途

此命令撤销对数据库的 Net Search Extender 更改。

权限

必须作为数据库管理员运行此命令才能禁用数据库。这要求您具有 DBADM 权限。

命令语法



connection-options:

```
|─┬──────────────────────────────────────────────────────┬──|
  └─CONNECT TO─database-name─┬────────────────────────────┬─┘
                             └─USER─userid─USING─password─┘
```

命令参数

CONNECT TO *database-name*

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户使用具有必要的 DB2 权限的用户标识运行该命令，那么可以省略此参数。

USER *userid* **USING** *password*

使用 *userid* 和 *password* 来连接至数据库。

FORCE

强制删除数据库中的所有 Net Search Extender 索引。

用法说明

此命令将复位已连接的数据库，以便其他 Net Search Extender 命令不再能够使用它。如果全文本索引存在于数据库中，那么此命令失败，除非使用 FORCE 选项。

此命令不会从 DB2 实例所有者除去 DBADM 权限。

注: 如果在数据库中定义了任何文本索引，那么禁用数据库将失败。建议依次除去这些索引，然后检查是否发生任何问题。如果将禁用数据库用于文本强制命令，那么它只保证除去数据库中的 Net Search Extender 目录表。

但是，如果一些索引不能彻底删除，那么可能仍然有一些资源需要手动清除。这些资源包括:

- 索引、工作和高速缓存目录中的文件
- ctedem.dat 中的调度程序条目
- 其中索引是使用复制捕获选项创建的，而远程数据库的表中的 IBMSNAP_SIGNAL、IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCNTL 条目必须手动删除。使用 APPLY_QUAL="NSE"||<instance name> and TARGET_SERVER= <database name> 命令很容易标识这些条目。

  在以下示例中，实例是 DB2，而数据库是 SAMPLE。

```
DELETE FROM <ccSchema>.IBMSNAP_SIGNAL
WHERE SIGNAL_INPUT_IN IN
        (SELECT MAP_ID FROM <ccSchema>.IBMSNAP_PRUNCNTL
                WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE');

DELETE FROM <ccSchema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';

DELETE FROM <ccschema>.IBMSNAP_PRUNE_SET
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';
```

对数据库的更改

将删除下列在数据库中用来启用 Net Search Extender 的修改:

- 数据库中的 Net Search Extender 目录视图。

- 由 Net Search Extender 创建的所有数据库对象。

对文件系统和共享内存的更改

如果使用 FORCE 选项，将删除索引文件。

如果使用 FORCE 选项，将删除作为索引的任何已激活高速缓存的高速缓存。

DB2EXTHL 命令

用途

DB2EXTHL 命令将更改突出显示的 UDF 的输入参数的最大大小。

缺省情况下，突出显示 UDF 将最大大小为 100 KB 的文档作为输入并返回 200 KB CLOB。根据数据库中最大文档的大小，可以将输入值增大为最大大小 1 GB。

权限

必须作为数据库管理员运行此命令才能启用数据库。这要求您具有 SYSADM 权限以便能够将 DBADM 授予 DB2 实例所有者。

命令语法

```
▶▶──db2exthl──new-highlight-input-size──────────────────────────▶◀
```

命令参数

new-highlight-input-size

突出显示 UDF 的新结果大小（以千字节计）。这是一个小于 1048576 的正整数。

文本表所有者的管理命令

本节描述文本表所有者的管理命令的语法。

这些命令是 DB2TEXT 命令的子命令。这些命令允许表的所有者创建和处理表列的全文本索引。

命令	用途
第 108 页的『ACTIVATE CACHE 命令』	激活高速缓存以便可以执行使用存储过程的搜索操作
第 109 页的『ALTER INDEX 命令』	更改索引的特征
第 113 页的『CLEAR EVENTS 命令』	从更新索引期间所使用的索引事件表中删除索引事件
第 114 页的『CREATE INDEX 命令』	创建全文本索引
第 125 页的『DEACTIVATE CACHE 命令』	取消激活高速缓存以使使用存储过程的搜索操作不再可行
第 128 页的『DB2EXTTH 命令』	编译同义字定义文件

命令	用途
第 127 页的『DROP INDEX 命令』	删除文本列的全文本索引
第 104 页的『ENABLE DATABASE 命令』	启用当前数据库以创建全文本索引
第 129 页的『UPDATE INDEX 命令』	启动基于文本列的当前内容的建立索引过程
第 131 页的『HELP 命令』	显示 DB2TEXT 命令选项的列表
第 132 页的『COPYRIGHT 命令』	显示 Net Search Extender 产品和版权信息

提示:

如果没有将任何数据库连接信息指定为 db2text 命令的一部分，那么 db2text 可执行文件将会建立与在环境变量 DB2DBDFT 中指定的缺省子系统的隐式连接。

ACTIVATE CACHE 命令

用途

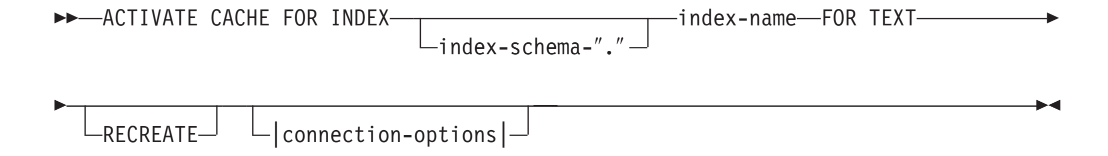
此命令将激活 DB2 用户表或持久高速缓存文件中已高速缓存的表。在完成之后，就可以执行使用存储过程的搜索操作。

仅当索引是使用 CACHE TABLE 选项创建的时候此命令才可用。

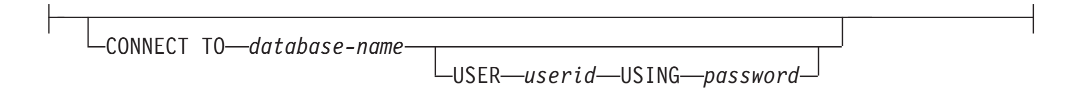
权限

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。

命令语法



connection-options:



命令参数

index-schema
: CREATE INDEX 命令中所指定的文本索引的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识。

index-name
: CREATE INDEX 命令中所指定的文本索引的名称。

RECREATE
仅适用于使用持久高速缓存的索引；会删除现有高速缓存。如果完成了没有激活的更新，那么会从数据库中自动重构持久高速缓存。

CONNECT TO *database-name*
作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

如果当前正在对索引运行下面的其中一个命令，那么不能发出该命令:

- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- CLEAR EVENTS
- DEACTIVATE CACHE

注: 激活高速缓存的表可能需要从头开始重新创建它，即使使用的是持久高速缓存也是这样。如果在取消激活持久高速缓存的同时执行了更新操作，就会发生这种情况。

用来构建高速缓存的内存量是根据当前文档数和结果列大小来动态计算的。使用 PCTFREE 值来将计算的最小内存量乘以因子 100/(100-PCTFREE)。PCTFREE 值是在 CREATE INDEX 或 ALTER INDEX 命令中指定的。

因此，PCTFREE 描述激活了高速缓存时保留给插入操作的已分配高速缓存的百分比。注意，对于每个 ACTIVATE CACHE 命令，都会重新估计实际内存大小。

对文件系统的更改
创建了用于实现持久高速缓存的文件。

ALTER INDEX 命令

用途

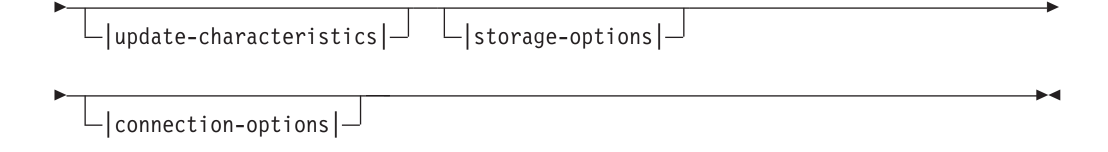
该命令将更改全文本索引的特征，例如，更新选项和存储器选项。

权限

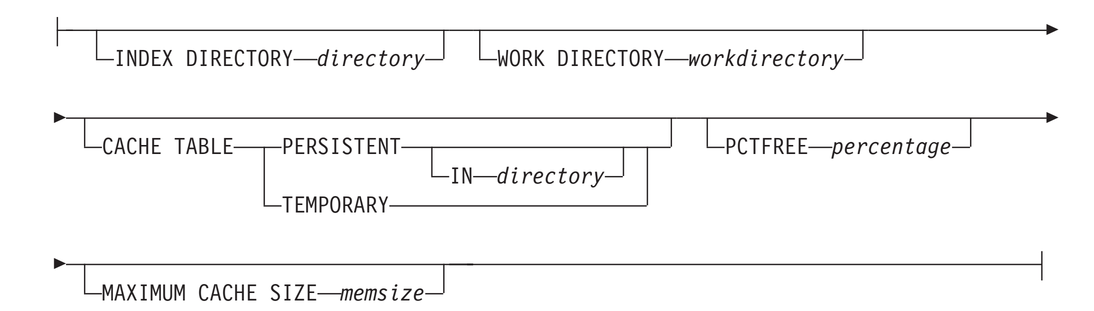
根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。

命令语法

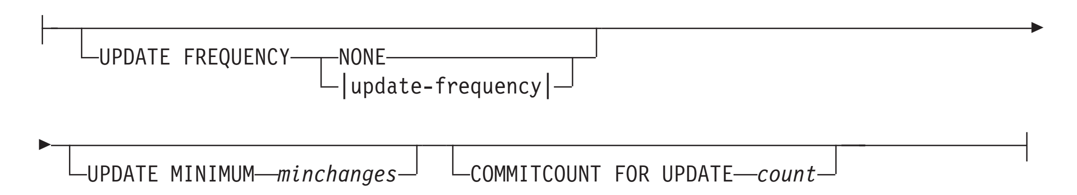




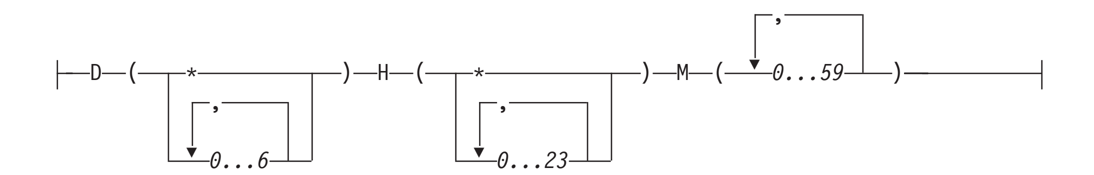
storage-options:



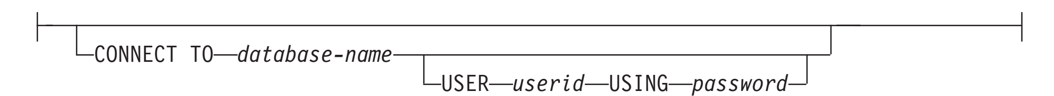
update-characteristics:



update-frequency:



connection-options:



命令参数

index-schema
: CREATE INDEX 命令中所指定的文本索引的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识。

index-name
: CREATE INDEX 命令中所指定的文本索引的名称。

INDEX DIRECTORY *directory*

用来存储文本索引的目录路径。由于该目录将包含索引数据，因此应确保 DB2 实例所有者用户标识对该目录具有读、写和执行许可权。

注意，在分布式 DB2 环境中，每个节点上都必须存在此目录。在目录下面创建一个子目录 `NODE<nr>`，以区分服务器的逻辑节点上的索引。将删除先前索引目录中的任何索引文件。

WORK DIRECTORY *workdirectory*

在搜索和管理操作期间用来存储临时文件。可以独立地更改新的索引目录的独立工作目录。

如果该目录不存在，那么会使用 DB2 实例所有者用户标识将它创建。如果该目录存在，那么确保在 UNIX 平台上实例所有者对该目录具有读、写和执行许可权。

注意，在分布式 DB2 环境中，每个节点上都必须存在此目录。在目录下面创建一个子目录 `NODE<nr>`，以区分服务器的逻辑节点上的索引。将删除先前索引目录中的任何临时索引文件。

CACHE TABLE PERSISTENT IN *directory*

指定即使在取消激活或系统重新引导之后，CREATE INDEX 中已高速缓存的表也是持久的。在任何一种情况下，这都允许快速执行 ACTIVATE CACHE。持久高速缓存存储在指定的目录中。

先前创建的持久高速缓存将移至新位置。此操作总是需要已取消激活的索引。

CACHE TABLE TEMPORARY

指定高速缓存的结果表现在是临时的，并且已经删除了任何先前存在的持久高速缓存。注意，此更改操作需要已取消激活的索引。

MAXIMUM CACHE SIZE *memsize*

指定在 ACTIVATE CACHE 期间要构建的高速缓存表的新的最大大小。以兆字节为单位将 memsize 参数指定为正整数。

如果 *memsize* 太小，那么 ACTIVATE CACHE 命令将失败。实际高速缓存大小是在执行 ACTIVATE CACHE 命令期间计算的。此更改需要已取消激活的索引。

PCTFREE *percentage*

指定保留的用于附加文档的高速缓存的百分比。该百分比必须是小于 100 并且大于或等于 0 的整数。注意，会删除先前的持久高速缓存，并且此更改需要已取消激活的索引。

UPDATE FREQUENCY

通过使用下列参数，索引更新频率可确定何时进行更新:

- `D`. 在星期几更新索引: *（每天）或 0..6（0 = 星期天）
- `H`. 在几点钟更新索引: *（每小时）或 0..23
- `M`. 在多少分钟更新索引: 0..59
- `NONE`. 不再进一步更新索引。对于不再进一步更改的文本列，或者将来只运行手动更新，就可以使用此参数。

如果不指定 UPDATE FREQUENCY 关键字，那么频率设置将保持不变。

UPDATE MINIMUM *minchanges*
在对索引进行增量更新之前允许对文本文档所作的最小更改数。如果不指定 UPDATE MINIMUM 关键字，那么不会更改设置。

注意，仅当索引不是使用 RECREATE ON UPDATE 选项创建时才能更改 UPDATE MINIMUM。

COMMITCOUNT FOR UPDATE *count*
对于更新处理，可以指定落实计数。这同时适用于 UPDATE 命令和 UPDATE FREQUENCY 规范，它将调度更新处理。

在初始更新期间将忽略 COMMITCOUNT 值。

注意，仅当索引不是使用 RECREATE ON UPDATE 选项创建时才能更改 COMMITCOUNT。

还要注意，如果是使用 REPLICATION 子句来创建索引的，那么不能更改 COMMITCOUNT。

CONNECT TO *database-name*
作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

如果正在对索引运行下面的其中一个命令，那么不能发出 Alter Index 命令:

- ALTER INDEX
- CLEAR EVENTS
- ACTIVATE CACHE
- DROP INDEX
- UPDATE INDEX
- DEACTIVATE CACHE

如果使用高速缓存选项来创建索引，那么当激活了索引时不能对索引目录使用 ALTER INDEX 命令。必须首先取消激活高速缓存。

在分布式 DB2 环境中，仅在具有单个节点的表空间上才允许具有高速缓存选项的文本索引。

对数据库的更改

更改 Net Search Extender 目录视图。

对文件系统的更改

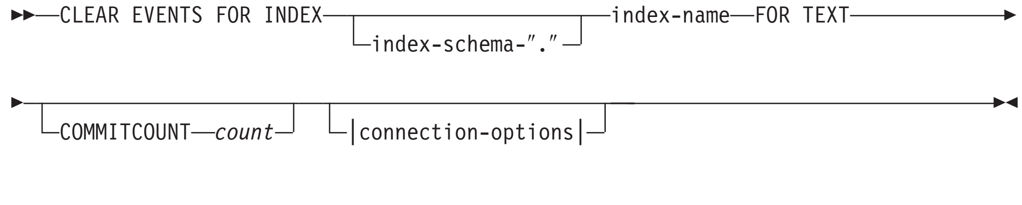
- 在索引中创建 `NODE<nr>` 子目录和工作目录
- 移动索引文件
- 创建持久高速缓存目录
- 移动持久高速缓存文件

CLEAR EVENTS 命令

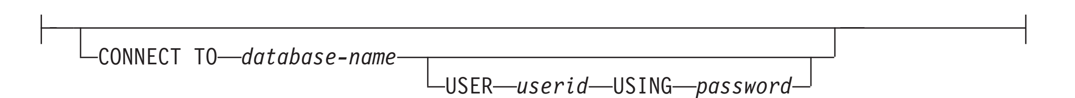
用途

此命令将从索引的事件视图中删除建立索引事件。可在 DB2EXT.TEXTINDEXES 视图的 EVENTVIEWNAME 列中找到事件视图的名称。

命令语法



connection-options:



命令参数

index-schema
: CREATE INDEX 命令中所指定的文本索引的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识。

index-name
: CREATE INDEX 命令中所指定的文本索引的名称。

COMMITCOUNT *count*
: 大于等于 0 的 INTEGER 值表示由 DB2 在一个事务中删除的行数。

CONNECT TO *database-name*
: 作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
: 使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

当在 CREATE INDEX 或 ALTER INDEX 命令中使用 UPDATE FREQUENCY 选项来调度常规更新时，就会定期检查事件表。在检查了发生错误的任何事件的原因并且除去了事件表中提到的错误源之后，使用 CLEAR EVENTS 来清除事件表。

如果正在对索引运行下面的其中一个命令，那么不能发出 Clear Events 命令:

- UPDATE INDEX
- ALTER INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE

• DROP INDEX

CREATE INDEX 命令

此命令对文本列创建全文本索引以供在 Net Search Extender 全文本查询中使用。

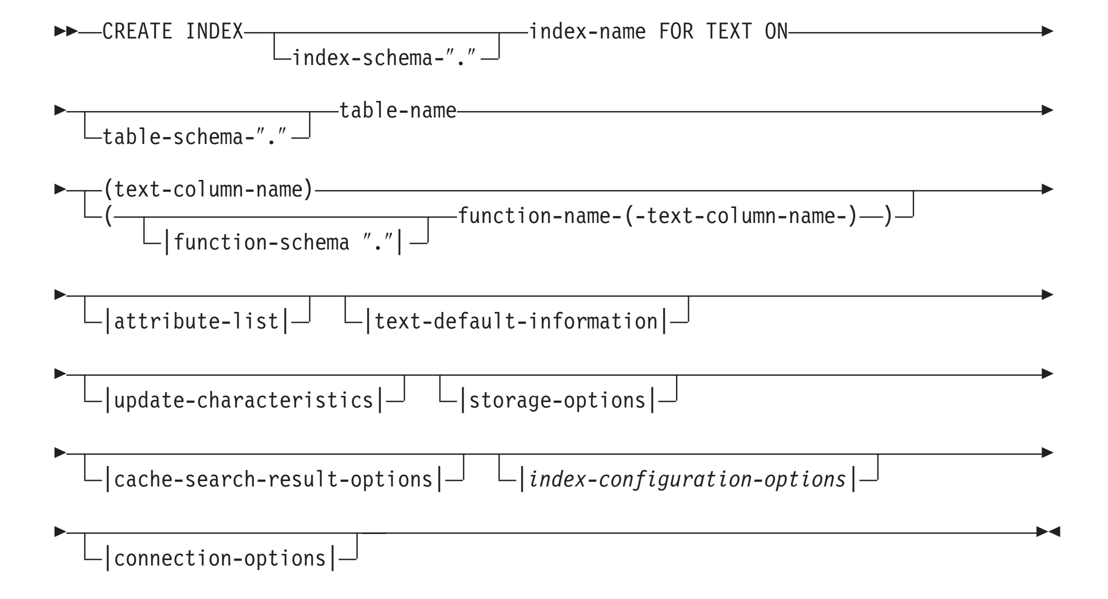
用途

在分布式 DB2 数据库环境下，全文本索引创建在表空间的每个分区中，用户表定义在该表空间上。不允许对表空间的分布进行后续更改，这将导致在执行管理命令和搜索过程中产生意外行为。

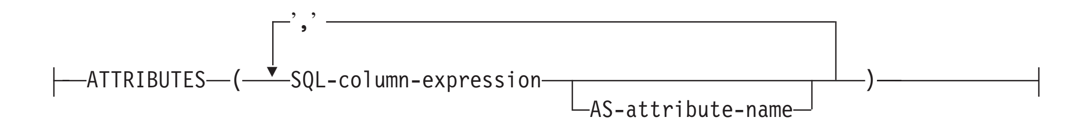
权限

根据 DB2 目录视图，此命令中的用户标识必须对在其中创建全文本索引的表具有 CONTROL 特权。

命令语法

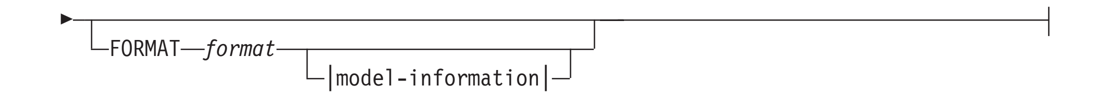


attribute list:

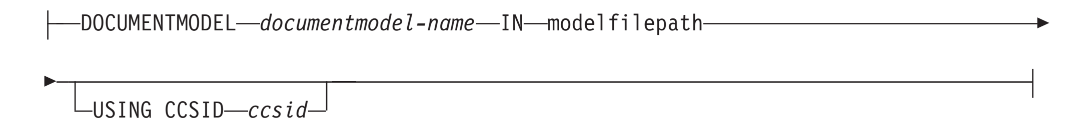


text-default-information:

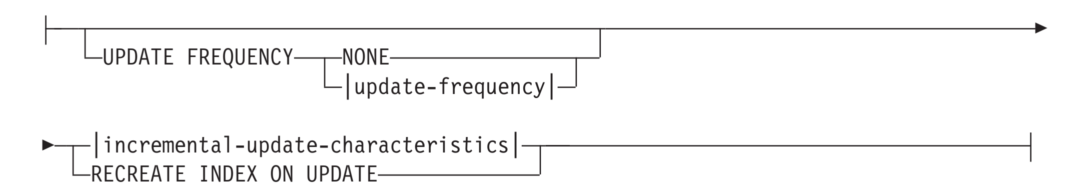
CCSID—*ccsid* LANGUAGE—*language*



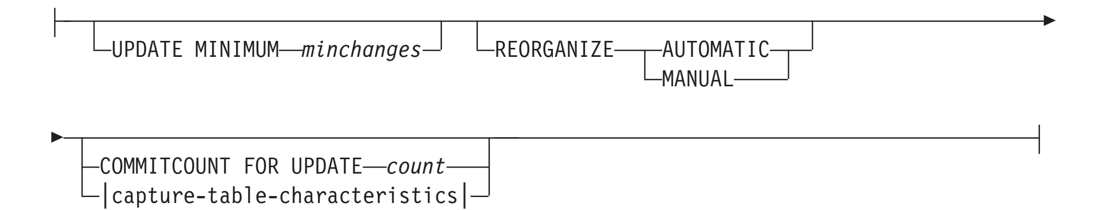
model-information:



update-characteristics:



incremental-update-characteristics:

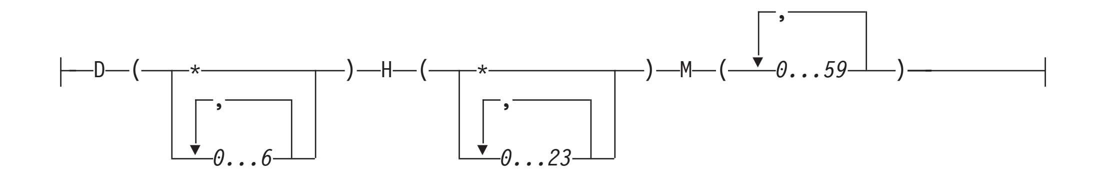


capture-table-characteristics:

```
├──REPLICATION CAPTURE TABLE──┬─────────────────────────────┬──capture-table-name──────►
                              └─capture-table-schema-"."──┘

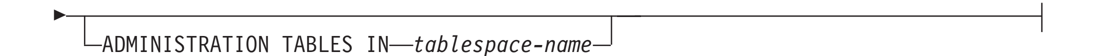
►─CONTROL TABLE SCHEMA──capture-control-schema─────────────────────────────────────────┤
```

update-frequency:

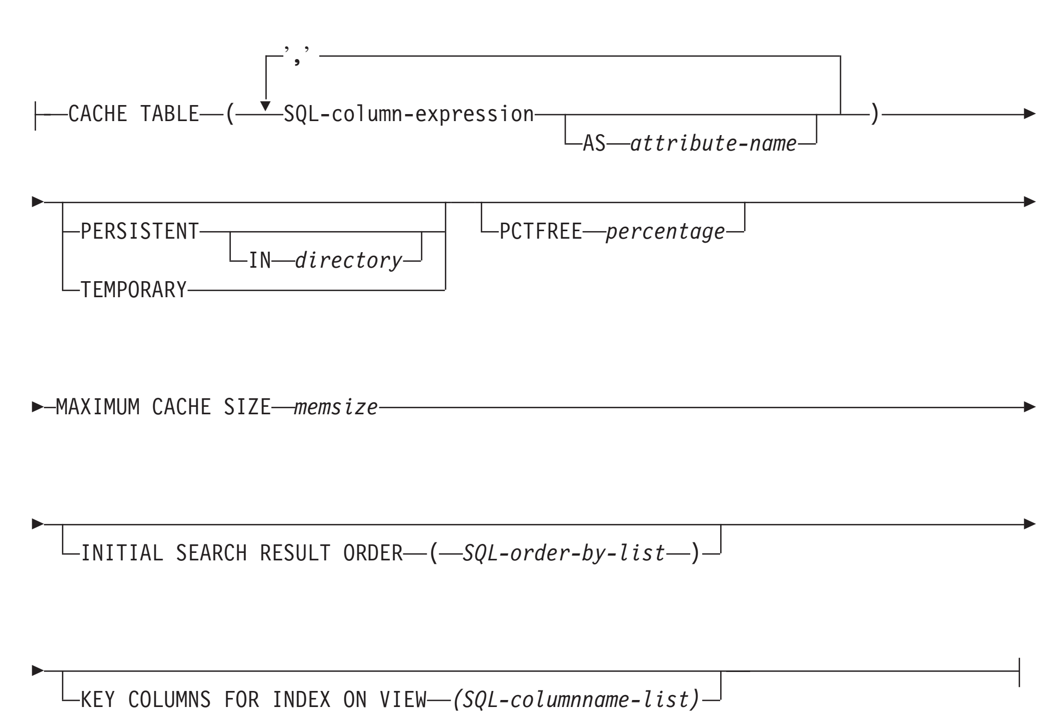


storage-options:

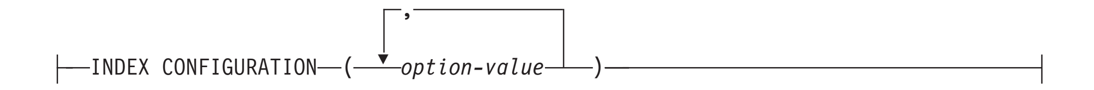




cache-search-results-options:



index-configuration-options:



connection-options:



命令参数

index-schema

文本索引的模式。将该模式用作特定于索引的管理表的 DB2 模式名。如果没有指定任何模式，那么使用 DB2 连接的用户标识。注意，索引模式必须是有效的 DB2 模式名。

index-name

索引的名称。它与索引模式一起用来唯一地标识数据库中的全文本索引。

注意，索引名必须是有效的 DB2 索引名。

table-schema

为其创建索引的表、昵称或视图的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识。

table-name

在已连接的数据库中包含为其创建了全文本索引的列的文本表、昵称或视图的名称。

注意，当表名不是指 DB2 基本表时，具有下列限制:

- 视图只允许进行存储过程或表值函数搜索。因此，必须使用 KEY COLUMNS FOR INDEX ON VIEW 子句指定索引或视图的键列。
- 对于对不具有捕获表的昵称的增量索引更新，会创建一个日志表。如果对昵称表或视图中的数据进行任何更改，那么必须手动填写日志表。对于基本表，此操作是自动完成的，因此用户**不能**修改日志表。
- 只有基本表或昵称而不是视图的索引才允许 DB2 谓词 CONTAINS、SCORE 和 NUMBEROFMATCHES。
- 仅当您在命令中指定 cache-search-result 选项时才允许视图的索引。

text-column-name

包含用于创建全文本索引的文本的列名。一列必须具有下面的其中一种数据类型:

- CHAR（对于位数据）
- VARCHAR（对于位数据）
- LONG VARCHAR（对于位数据）
- CLOB
- DBCLOB
- BLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- XML

如果列数据类型不是以上任一数据类型，那么使用 ***function-schema.function-name*** 指定变换函数来将列类型转换为受支持的类型。

注意，允许存在相同列的若干个索引，但是，**只有**满足下列条件之一时才允许:

索引是对视图创建的

因此，不能在 CONTAINS、SCORE 或 NUMBEROFMATCHES 搜索自变量中使用索引。

索引是对表创建的

如果所有索引都同步，那么它们在下列 CREATE INDEX 命令详细信息中的同一列上具有完全相同的属性:

- Function name and schema
- ATTRIBUTES
- CCSID
- LANGUAGE

- FORMAT
- DOCUMENTMODEL
- INDEX CONFIGURATION

因此，CONTAINS、SCORE 或 NUMBEROFMATCHES 自变量选择哪个索引并不重要。

function-schema.function-name
用来访问在具有不受支持类型的一列中的文本文档的用户定义的函数的模式和名称。该函数通过将任意列类型用作输入参数来执行列类型转换。它返回的值是 Net Search Extender 支持的其中一种类型。

ATTRIBUTES (*SQL-column-expression* **AS** *Attribute-name***, ...)**
确保除了文本列之外还对列表达式的内容建立了索引。然后可以通过搜索语句中的 ATTRIBUTES 子句来搜索此内容。必须使用对其创建索引的表的非限定列名来定义 SQL 列表达式。唯一允许的数据类型为 DOUBLE。可以在列表达式中使用强制类型转换运算符，但是，**不可能**进行 DB2 的隐式数据类型转换。属性名必须遵循文档模型中的属性名的规则，并且必须与索引模型定义文件中的所有属性名不相同。

通过使用下列规则来确定表达式的属性名:

- 如果在列表达式中由 SQL AS 子句显式命名，那么使用指定的名称。以下是一个示例:

```
ATTRIBUTES (C1+C2 AS myname)
```

- 如果使用指定表的列时不带 AS，那么使用该列的名称。例如:

```
ATTRIBUTES (C1)
```

- 如果使用表达式时不带 AS，并且表达式不引用已命名的列，那么 CREATE INDEX 会报告错误。

例如:

```
ATTRIBUTES (CAST(JULIAN_DAY(date) AS DOUBLE) as day, (price1+price2)/2 as avg_price)
```

注意，未用单引号引起来的属性会被映射为大写，并且在搜索期间必须用大写字母指定这些属性。

CCSID *ccsid*
当对文本文档建立索引时，使用“编码字符集标识”。缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='CCSID'。

仅当一列的数据类型为二进制时才设置 CCSID。

LANGUAGE *language*
language 参数指定在索引配置值 IndexStopWords 设置为 0 的情况下（在此情况下，建立索引期间将忽略停用词）所选择的停用词字典使用的语言。对于泰国语（TH_TH），必须总是设置此参数，以启用“泰国语”单词断行；对“Turkish”设置此参数以正确区分带点的“i”和不带点的“i”的使用。

FORMAT *format*
一列中的文本文档的格式，例如，HTML。此信息对于为文档建立索引是必需的。

对于结构化文档格式，可以在文档模型文件中指定信息。如果没有指定任何文档模型，那么使用缺省文档模型来对文档的文本建立索引。

如果没有指定 format 关键字，那么缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='FORMAT'。Net Search Extender 设置的初始缺省值为 TEXT。对于数据类型 XML，在没有指定格式的情况下，格式被缺省设置为 XML。如果数据类型是 XML，那么允许您为 FORMAT 指定的唯一值是 XML。

DOCUMENTMODEL *documentmodel-name* **IN** *modelfilepath*

modelfilepath 指定模型文件的位置。modelfilepath 必须是一个标准路径。模型文件包含 FORMAT 子句中的格式的模型定义。DB2 实例所有者必须可以将它读取。文档模型使您能够对文档的特定部分建立索引和进行搜索。可以在文档模型中定义标记（markup tag）和节名。将文档模型绑定至支持 HTML、XML 或 GPP 结构的文档格式。在一个模型文件中只能指定一种文档模型。

注意，因为只有在执行 CREATE INDEX 命令期间才读取文档模型，所以，不能识别此索引的任何后续更改。

注意，在分布式 DB2 环境中，必须使用共享文件系统来确保在每个节点上都能够访问 `modelfilepath`。

USING CCSID *ccsid*

指定 CCSID 来解释模型文件的内容。缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='MODELCCSID'。

UPDATE FREQUENCY

索引更新频率确定何时进行更新。如果对用户表的更改数小于由 UPDATE MINIMUM 选项指定的更改数，那么不会更新索引。如果不指定 UPDATE FREQUENCY，那么使用缺省值 NONE，以便不更新索引。当您不想再更改文本列或者想手动控制更新过程时，此选项是很有用的。

- **D.** 在星期几更新索引：*（每天）或 0..6（0 = 星期天）
- **H.** 在几点钟更新索引：*（每小时）或 0..23
- **M.** 在多少分钟更新索引：0..59
- **NONE.** 不再进一步更新索引。必须手动启动更新。

缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='UPDATEFREQUENCY'。

如果您决定不使用 UPDATE FREQUENCY 参数来安排自动更新索引，那么可以改为使用操作系统函数（例如，crontab）。

UPDATE MINIMUM *minchanges*

在根据 UPDATE FREQUENCY 设置来更新索引之前需要对文本文档进行的最小更改数。只允许使用正整数值。缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='UPDATEMINIMUM'。

注意，如果手动执行 DB2TEXT UPDATE 命令，那么会忽略此值。此选项不能与 RECREATE INDEX ON UPDATE 选项配合使用，原因是没有日志表和增量更新的触发器，更改数不可用。

对于分布式数据库，对每个节点检查 UPDATE MINIMUM。

REORGANIZE AUTOMATIC | MANUAL

如果指定了 REORGANIZE AUTOMATIC 时，那么根据更新频率设置执行的更新将只识别索引。此步骤是在更新之后根据 `select REORGSUGGESTED from DB2EXT.TEXTINDEXES` 的值自动完成的。

只能使用手动 UPDATE 命令并通过使用 REORGANIZE 选项来执行 REORGANIZE MANUAL。

如果省略 REORGANIZE 子句，那么缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='AUTOMATICREORG'。

REPLICATION CAPTURE TABLE *capture-table-schema***.***capture-table-name* **CONTROL TABLE SCHEMA** *capture-control-schema*

对于增量更新处理，采用指定的复制捕获表而不是通常为索引创建的日志表。因此，`schemaname`、`tablename` 和复制捕获表名与本地 DB2（联合）数据库中的对象相关。

capture-control-schema 是复制控制表的模式名，例如，本地 DB2 上的 IBMSNAP_PRUNE_SET。在设置了复制之后，复制控制表必须可用作本地 DB2 系统上的昵称。

至少，必须具有可用于下列捕获控制表的昵称:

- IBMSNAP_SIGNAL
- IBMSNAP_PRUNE_SET
- IBMSNAP_PRUNCNTL
- IBMSNAP_REGISTER
- IBMSNAP_REG_SYNC（仅适用于非 DB2 远程源）

由于 DB2 复制中心不会自动保证为远程捕获表和捕获控制表创建本地昵称，因此，可能需要手动创建。该任务类似于为要对其创建文本索引的表创建昵称。

用户表昵称与捕获表昵称中的主键列的列名必须匹配。另外，不能更改捕获表昵称中的 IBMSNAP_OPERATION、IBMSNAP_COMMITSEQ 和 IBMSNAP_INTENTSEQ 列的名称。

创建索引之后，列名 DB2EXT.TEXTINDEXES(LOGVIEWNAME) 和 DB2EXT.TEXTINDEXES(LOGVIEWSCHEMA) 都表示复制捕获表的局部名。

由于 Net Search Extender 并不需要 DB2 复制中心的所有功能，所以“更改数据”（CD）表或“一致更改数据”（CCD）表必须遵守下列规则:

- 使用更改捕获注册但不使用完全刷新复制选项。
- 不允许捕获更改的任何水平子集。例如，通过触发器。请参阅《DB2 复制指南和参考，版本 8》的第 6 章『在复制环境中对数据划分子集』。
- 仅当包括主键列、文本列和 DB2TEXT CREATE INDEX 命令的属性和高速缓存表表达式中涉及到的所有列时，才允许注册列子集的更改。
- 主键列必须包括在捕获表中。注意，后映像是足够的。
- 不能压缩捕获表。对于每个主键，必须具有一个条目具有最新数据。但是，Net Search Extender 要求所有历史记录都可用。
- 表必须使用 D/I 选项。这使得能够将对源表上的主键的更新变换成插入/删除对。

其他先决条件包括:

- 对其创建索引的源表的服务器类型和版本是下列各项的其中一项:
  - 在 AIX 操作系统上运行的 DB2 版本 9.1 或更高版本
  - 在 HP-UX 操作系统上运行的 DB2 版本 9.1 或更高版本
  - 在 Linux 操作系统上运行的 DB2 版本 9.1 或更高版本

- 在 Solaris 操作系统上运行的 DB2 版本 9.1 或更高版本
  - Informix® IDS 9.3
  - ORACLE 9i
  - SYBASE ASE 12.5
  - Microsoft SQL Server 2000
- 以下是受支持的包装器的列表:
  - DB2: DRDA®
  - Informix: Informix
  - ORACLE: NET8 和（SQLNET）
  - SYBASE: CTLIB
  - MSSQLSERVER: MSSQLODBC3

注: 确保将正确的源表名插入到注册表中。视远程 DBMS 的类型不同，必须使用远程表名或本地昵称:

- DB2: 远程表名（远程服务器上的表名）
- 非 DB2: 本地昵称（联合 DB2 数据库中的相应昵称）

必须存在用户映射本地用户才能通过昵称来访问远程数据源，而远程用户必须对表具有 CONTROL 特权。

如果 DB2 实例所有者用户标识不同于本地用户标识，那么需要 DB2 实例所有者用户标识的附加用户映射。

指定的基本表名不能是昵称的视图。这是因为视图可建立在若干个昵称基础之上，并且还可能涉及到若干个 CD 表和 CCD 表。由于在复制捕获子句中只能指定一个 CD 表或 CCD 表，所以不能支持昵称的视图。另外，不能支持远程视图的昵称，这是因为缺少主键。

CD 或 CCD 表必须是昵称而不能是视图或别名。

COMMITCOUNT FOR UPDATE *count*

对于**增量**更新处理，可以指定落实计数。如果未指定它，那么缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='COMMITCOUNT'。

可以在 DB2EXT.TEXTINDEXES.COMMITCOUNT 中找到索引的 COMMITCOUNT FOR UPDATE 值。可以使用 ALTER INDEX 命令更改每个索引的这个值。它还适用于根据 UPDATE FREQUENCY 规范的已调度更新处理。如果值为 0，那么意味着更新是在一个事务中完成的；如果值大于 0，那么指定要在一个事务中处理的文档数。建议您不要使用非零 COMMITCOUNT。如果您必须使用非零 COMMITCOUNT，那么应将它设置为一个足够大的值，以使单个增量更新期间进行的立即落实数保持非常小。每次落实时都会移动索引文件和清除索引日志文件，完成此过程将花费大量时间，重复执行此操作时更是如此。

如果未设置 COMMITCOUNT，那么不会更新 db2ext.textindexes 中的 NUMBER_DOCS 参数。因此，要在更新过程中查看已经处理的文档数，使用 CONTROL LIST 命令。

RECREATE INDEX ON UPDATE

这不允许增量索引更新，但是会在（由命令或已调度的更新）执行更新操作时重新创建索引。

注: 如果指定了 RECREATE INDEX ON UPDATE，那么不会对用户表创建任何触发器，也不会创建日志表。

INDEX DIRECTORY *directory*

要用来存储文本索引的目录路径。由于该目录将包含索引数据，因此应确保 DB2 实例所有者用户标识对该目录具有读/写和执行许可权。

缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME=INDEXDIRECTORY'。在目录下面创建一个子目录 NODE<nr>，以区分服务器的逻辑节点上的索引。

注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

WORK DIRECTORY *directory*

（可选）可以指定独立的工作目录，该目录将在执行索引搜索和管理操作期间用来存储临时文件。该目录必须存在，并且对于 DB2 实例所有者用户标识都具有读/写和执行许可权。

缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='WORKDIRECTORY'。在目录下面创建一个子目录 NODE<nr>，以区分服务器的逻辑节点上的索引。

注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

如果未指定 WORK DIRECTORY，那么会在 INDEX DIRECTORY 中创建一个名为 work 的目录。

注: 在 INDEX DIRECTORY 和 WORK DIRECTORY 中创建的文件将遵循为实例所有者设置的 umask 限制。确保这些 umask 限制必须允许不受防护用户的组读/写访问权。

ADMINISTRATION TABLES IN *tablespace-name*

为索引创建的管理表的常规表空间的名称。表空间必须存在。如果不指定此项，那么会选择用户表的表空间（如果索引是对基本表创建的）。

对于昵称或视图，DB2 将选择缺省表空间。

当为了在分布式 DB2 环境中进行存储过程搜索而对视图、昵称或文本索引创建文本索引时，必须在单个节点上指定用于管理表的表空间并且必须在此节点上显式调用该表空间。

为了确保连接至正确的节点，使用 DB2NODE 环境变量。

CACHE TABLE (*SQL-column-expression-list***)**

除了索引之外，还会构建高速缓存表，该表由指定的列表达式组成。此高速缓存用来通过存储过程搜索返回结果集而不必将全文本搜索结果与 DB2 表连接起来。注意，始终可以执行将全文本索引与 CONTAINS 函数配合使用的常规 DB2 搜索。

使用对其创建索引的表的非限定列名来定义 SQL 列表达式。允许的 SQL 列表达式类型是所有内置的和用户定义的单值类型。结果集中的列名是使用下列规则确定的:

- 如果在列表达式中由 SQL AS 子句显式命名，那么使用指定的名称。例如:
```
CACHE TABLE (C1+C2 AS myname)
```
- 如果使用指定表的列时不带 AS 子句，那么使用该列的名称。例如:

CACHE TABLE(C1)

- 如果使用表达式时不带 AS，并且表达式不引用已命名的列，那么 CREATE INDEX 会报告错误。
- 不允许重复的列名。

不支持将 CLOB 数据类型作为高速缓存数据类型。需要将这些数据类型强制类型转换为 VARCHAR。

> **注**：注意，如果结果集的列名不是互不相同的，那么 CREATE INDEX 命令会返回错误。还要注意，在创建之后，高速缓存表不是隐式激活的，例如，在执行 DB2TEXT ACTIVATE CACHE 之前，不能进行由存储过程执行的搜索。

仅当用户表存储在具有单个节点的表空间中时，才能在分布式 DB2 环境中使用此选项。

PERSISTENT IN *directory*

指定也会持久创建高速缓存。在取消激活持久高速缓存或者系统重新启动之后，重新激活持久高速缓存比重新激活非持久高速缓存的速度更快。持久高速缓存存储在指定的目录中。

注意，如果没有指定目录，那么缺省值取自 db2ext.dbdefaults 视图，其中 DEFAULTNAME='CACHEDIRECTORY'。

TEMPORARY

指定高速缓存不是持久存储的。如果既没有指定 PERSISTENT，也没有指定 TEMPORARY，那么缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='USEPERSISTENTCACHE'。

MAXIMUM CACHE SIZE *memsize*

指定在 DB2TEXT ACTIVATE CACHE 期间要构建的高速缓存表的最大大小。必须以兆字节为单位将 memsize 参数指定为正整数。没有为 memsize 指定缺省值。如果该整数太小，那么 ACTIVATE CACHE 命令将失败。实际高速缓存大小是在执行 ACTIVATE CACHE 命令期间计算的。

在各种 32 位平台上，最大高速缓存大小的限制分别为:

- Windows: 1024 MB（1 GB = 1073741824 字节）
- Linux: 2048 MB（2 GB = 2147483647 字节）

在 64 位安装上，最大高速缓存大小限制取决于可用内存量。

PCTFREE *percentage*

指定要为附加文档保留的高速缓存的百分比。该百分比必须是小于 100 并且大于或等于 0 的整数。如果未指定此项，那么缺省值取自 db2ext.dbdefaults 视图，其中 DEFAULTNAME='PCTFREE'。缺省值为 50%。

INITIAL SEARCH RESULT ORDER (*SQL-order-by-list***)**

指定在初次建立索引期间用于检索用户表内容的顺序。当使用此选项时，通过跳过对全文本搜索结果进行动态排序，文档就会像存储在高速缓存结果表中一样按它们的建立索引顺序返回。

仅当您正在使用“存储过程”搜索界面时，才能使用预先排序的索引和预定义的搜索结果排序。例如: INITIAL RESULT ORDER(length(column1) asc, column2+column3 desc)。

对于 SQL 标量搜索函数和表值函数未实现预定义的搜索结果排序。

注: 对于增量更新之后的新文档或已更改的文档，**不能**确保该索引顺序。

KEY COLUMNS FOR INDEX ON VIEW (*SQL-columnname-list***)**

如果对视图创建了索引，那么必须指定 KEY COLUMNS FOR INDEX ON VIEW 子句，否则，“不能”指定该子句。列名列表指定“唯一地”标识视图中的某行的各列。

由于 DB2 不能对主键检查这种唯一性，因此，用户应负责确保等价的唯一性。指定的列为索引构建日志表的一部分。

INDEX CONFIGURATION (*option-value***), ...**

这些是索引配置值。缺省值加上了下划线。

选项	Values	描述
TreatNumbersAsWords	0 或 1	如果设置为 1，即使一系列数字与字符相邻，也会将这一系列数字解释为独立的单词。缺省值为 0，它意味着 tea42at5（这是一个示例）会被认为是一个单词。
IndexStopWords	0 或 1	在建立索引期间忽略或者考虑停用词。缺省值为 1，它将对所有文本（包含停用词）创建索引。目前，停用词列表是 `<instance>/sqllib/db2ext/resources` 目录中的一个 UCS-2 文件 `<language>.tsw`。创建索引之后，更改此文件**没有任何影响**。还要注意，`<language>` 是 CREATE INDEX 命令中的 LANGUAGE 值。
UpdateDelay	秒	指定不使用捕获表的增量更新的持续时间（以秒计）。将仅从日志表中抽取早于此持续时间的条目。这是为了避免丢失更新。例如，在用户事务对更新命令有影响的事务情况中，未反映在索引中的文档更改。因此，UpdateDelay 参数应设置为对其创建索引的表的用户写事务的最长持续时间。
IgnoreEmptyDocs	0 或 1	如果 IgnoreEmptyDocs 设置为 1，那么在索引中不表示空文档（内容长度为 0 或空值）。如果使用此选项并且文档内容为空，那么下一个增量更新将从索引中删除这些文档。

CONNECT TO *database-name*

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*

使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

对数据库的更改

- 更改 Net Search Extender 目录视图。
- 在指定的表空间中创建索引日志表。仅当未指定 RECREATE INDEX 选项并且未指定捕获表时才执行此操作。
- 在指定的表空间中创建索引事件表。

- 因循第一次更新：对用户文本表创建触发器（仅当未指定 RECREATE INDEX 并且未使用捕获表时）
- 如果使用了复制捕获表，那么会对捕获控制表进行下列更改：
  - 插入到 IBMSNAP_PRUNCTNL 和 IBMSNAP_PRUNE_SET 表中

  这些表中的条目是由下面的列唯一标识的：
  - APPLY_QUAL='NSE' || <DB2 instance running NSE>
  - SET_NAME= <internal index identifier>
  - TARGET_SERVER=<DB2 database name target to DB2TEXT operation>

对共享内存的更改

因循 ACTIVATE 执行：如果使用了 CACHE TABLE 子句，那么会在*共享内存*中构建结果表的高速缓存。

对文件系统的更改

- 在索引目录、工作目录和高速缓存目录下创建了 `NODE<nr>` 子目录。
- `<internal index name>` 目录是在 `<indexdirectory>/NODE<nr>` 下创建的，其中 `indexdirectory` 指的是此命令的相应参数，而 `NODE<nr>` 与分布式 DB2 环境中的节点号相关。

用法说明

创建全文本索引需要用户表的主键。在 DB2 Net Search Extender V9.1 中，可以使用包含多列的 DB2 主键，对类型没有任何限制。但是，要使用表值搜索，不允许任何复合主键。

主键列的数目最多为 62 列。对于页大小为 4k 的表空间，所有主键列的总长度最长为 1007 个字节；对于页大小为 8k 的表空间，最长为 2031 个字节；对于页大小为 16k 的表空间，最长为 4079 个字节；对于页大小为 32k 的表空间，最长为 4096 个字节。注意，如果主键中包含多列，那么上面所提到的长度限制应减小，每个附加列减少两个字节。

- ATTRIBUTES、CACHE TABLE 和 INITIAL SEARCH RESULT ORDER 的 SQL 表达式的总大小不能超过 24 KB。
- 初始索引更新总是作为一个逻辑事务来完成的，在这种情况下不存在落实计数。

注： 在创建索引之后，不能使用 ALTER TABLE 命令来更改主键列或视图键列的长度。

用户表、全文本索引和高速缓存的结果表之间的同步是在执行更新索引命令期间完成的。

DEACTIVATE CACHE 命令

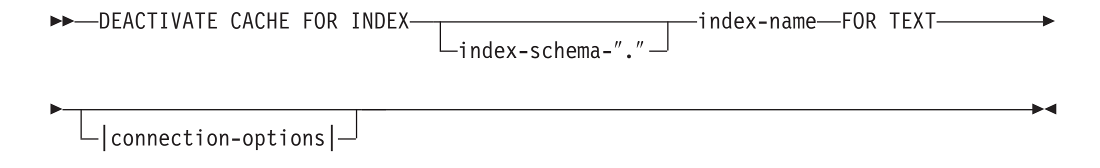
用途

此命令将释放高速缓存的表。保留持久高速缓存以便在下一个 ACTIVATE 命令中复用。在下一次激活之前，不再可以在取消激活的高速缓存中执行通过存储过程执行的搜索操作。

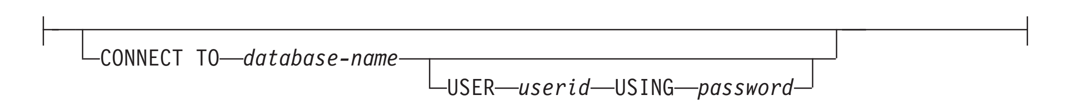
权限

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。

命令语法



connection-options:



命令参数

index-schema
: CREATE INDEX 命令中所指定的文本索引的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识作为模式名。

index-name
: CREATE INDEX 命令中所指定的文本索引的名称。

CONNECT TO *database-name*
: 作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
: 使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

注意，当正在对索引运行下面的其中一个命令时，不能发出此命令:

- ACTIVATE CACHE
- DEACTIVATE CACHE
- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- CLEAR EVENTS

注: 在取消激活持久高速缓存之后，通过存储过程执行的搜索就不能访问该高速缓存。但是，除非同时执行更新，否则，这可用于快速 ACTIVATE。

在这种情况下，将使用 ACTIVATE CACHE 命令自动从头开始重新创建持久高速缓存。

DROP INDEX 命令

用途

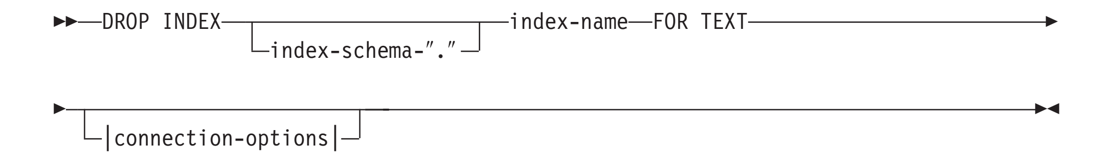
此命令将删除文本列的全文本索引。如果激活了索引的高速缓存，那么使用此命令来删除索引。

权限

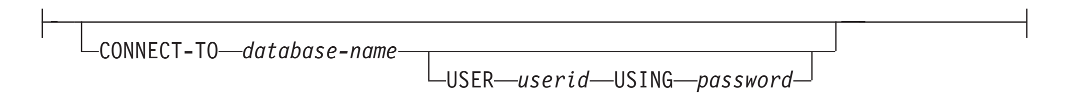
根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。或者，用户可以是数据库管理员（DBADM）。

或者，数据库管理员（DBADM）可以删除索引，原因是他们必须能够使用 FORCE 选项来禁用数据库。

命令语法



connection-options:



命令参数

index-schema
: CREATE INDEX 命令中所指定的文本索引的模式。如果没有指定任何模式，那么使用 DB2 连接的用户标识作为模式名。

index-name
: CREATE INDEX 命令中所指定的索引名称。借助索引模式，索引名将唯一地标识数据库中的全文本索引。

CONNECT TO *database-name*
: 作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
: 使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

无论索引的已高速缓存表的激活状态如何，都将删除索引。

注意，当正在对索引运行下面的其中一个命令时，不能发出该命令:
- UPDATE INDEX

- CLEAR EVENTS
- ALTER INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE
- DROP INDEX

注: 在废弃 DB2 中的用户表前后，必须手动删除索引。如果不手动删除索引，那么不能正确清理索引目录。

对数据库的更改

- 更改 Net Search Extender 目录视图
- 删除 DB2 索引
- 删除索引日志表/事件表
- 删除用户文本表的触发器

当使用复制捕获表时，会除去 IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCTRNL 表中的条目。

对共享内存的更改

删除了高速缓存的表。

对文件系统的更改

- 删除了已删除索引的索引目录和工作目录中的 `<internal index name>` 目录
- 删除索引的持久高速缓存

DB2EXTTH 命令

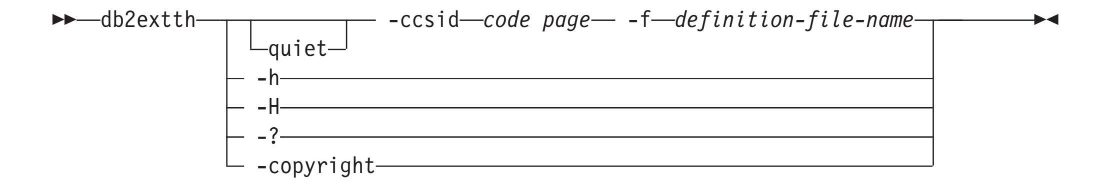
用途

这个独立的实用程序将编译同义字定义文件。运行了同义字编译器之后，就可以使用搜索自变量语法的与 THESAURUS 相关的功能部件。

权限

无。不需要对表所有者限制此命令，但是只在查询的上下文中有意义。

命令语法



命令参数

-f *definition-file-name*
: 包含同义字定义的文件的名称。该文件名必须包含文件的绝对路径或相对路径。该文件名最多为 8+3 个字符，扩展名是可选的。

同义字典是在定义文件所在的目录中生成的，并且它们名称相同。唯一的区别是字典具有下列扩展名：wdf、wdv、grf、grv、MEY、ROS、NEY、SOS 和 Ikn，其中 n 是一个数字。注意，如果现有同义字文件具有相同名称，那么会将它们覆盖。

-ccsid *code page*
编写同义字定义文件所使用的代码页。

-quiet 不显示输出信息。

-copyright
返回产品的内部构建号。当报告问题时使用此编号。

-h | -H | -?
显示帮助信息。

用法说明

使用此命令来将同义字定义文件编译为二进制同义字定义格式。

UPDATE INDEX 命令

用途

此命令通过显示最新索引来立即启动建立索引过程，以反映与索引相关联的文本列的当前内容。

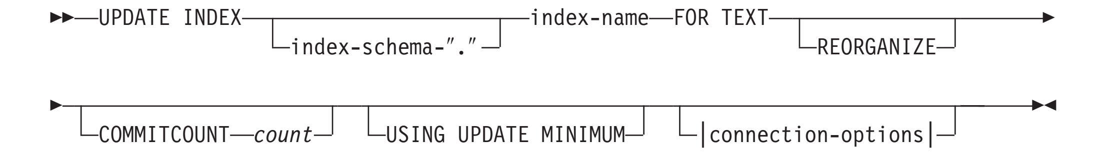
当正在执行更新时，可以执行使用 CONTAINS 谓词的搜索。对于具有已激活的高速缓存结果表的索引，在更新期间，也可以执行通过存储过程执行的搜索。但是，高速缓存表中的列可能会显示新值，即使尚未将已更改的文本落实到全文本索引中也是如此。

在 CREATE INDEX 命令中使用 RECREATE INDEX ON UPDATE 选项将在重新创建之前清除索引。在完成更新之前，将返回空结果。

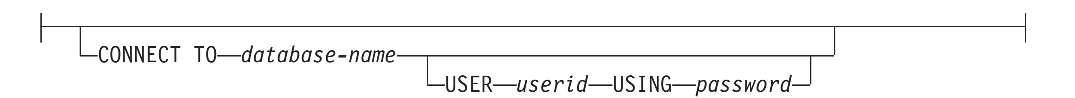
权限

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。

命令语法



connection-options:



命令参数

index-schema
: 文本索引的模式。这是在 CREATE INDEX 命令中指定的。如果没有指定任何模式，那么使用 DB2 连接的用户标识。

index-name
: 文本索引的名称。这是在 CREATE INDEX 命令中指定的。

REORGANIZE
: 如果频繁更新文本列，那么对索引的后续更新可能会变得不足。要使更新过程再次变得是足够的，可以重组索引。使用 DB2EXT.TEXTINDEXES 视图来确定是否需要重组索引。

  使用 CREATE INDEX 命令的 REORGANIZE AUTOMATIC 选项来避免手动检查和重组索引。

注: 重组过程是在常规更新之后进行的。

USING UPDATE MINIMUM
: 仅当达到了指定的更改数时才在 CREATE INDEX 命令中使用 UPDATE MINIMUM 设置并启动增量更新。缺省值是无条件启动更新。

  对于分布式数据库，对每个节点检查 UPDATE MINIMUM。

COMMITCOUNT *count*
: 大于等于 0 的 INTEGER 值确定由搜索引擎和由 DB2 在一个事务中为增量索引更新处理的文档数。可以使用 ALTER INDEX 命令来更改此值。

  但是，对于首次更新，例如，执行 CREATE INDEX 命令之后的第一次更新或者使用 RECREATE INDEX ON UPDATE 选项进行的任何更新，只存在一个逻辑事务，它忽略了 COMMITCOUNT。

  建议不要使用非零 COMMITCOUNT。

CONNECT TO *database-name*
: 作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，那么可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER *userid* **USING** *password*
: 使用 *userid* 和 *password* 来连接至数据库。如果未指定它们，那么会尝试使用当前用户标识而不带密码来建立连接。

用法说明

此命令以同步方式运行。它在分布式 DB2 环境中的所有必需的 DB2 逻辑/物理节点上启动更新处理。持续时间取决于要建立索引的文档数和已经建立索引的文档数。可以通过为每个索引创建的视图来查看更新的状态。可以从 DB2EXT.TEXTINDEXES 的 EVENTVIEWNAME 列中检索此视图的名称。

可以使用两个选项来查看已经处理的已落实文档的数目。要确定一个更新是否仍然在运行以及已经将多少个文档落实到索引中，使用 DB2EXT.TEXTINDEXES（NUMBER_DOCS）视图。使用与索引相关联的事件视图以获取有关启动、落实更改和完成更新处理的信息。

要查看已经处理的文档数，使用 CONTROL LIST ALL LOCKS FOR INDEX 命令。

注：这些视图只显示来自于已连接的节点的信息。

对于具有多个物理节点的基本表的增量更新，每个节点上的时间必须同步。如果时间不同步，那么更新可能丢失或根本不发生。

如果正在对索引运行下面的其中一个命令，那么不能发出该命令:

- CLEAR EVENTS
- ALTER INDEX
- DROP INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE
- UPDATE INDEX

在使用已取消激活的持久高速缓存结果表更新了索引之后，就会删除持久高速缓存，以便下一个 `ACTIVATE CACHE` 命令根据数据库内容重新将它创建。

如果用户中断此命令，那么更新功能涉及到的所有进程都会停止。如果在增量更新中使用了落实计数，那么可能已经落实了某些更新并且在索引中可以看见这些更新，而其他更新可能需要执行新的更新命令。强制中断更新索引过程可能会破坏索引。

要停止自动更新索引，在用于更新服务的分区上查找正在运行更新索引命令的 DB2 实例所有者进程。停止所有分区上的此进程和更新处理。

注：由于该命令在所有分区上的索引创建和初始索引更新的两个独立阶段起作用，因此，发出 `db2text drop index` 命令来确保索引不是部分可用。如果不发出此命令，那么可能由手动更新命令或更新频率选项触发的下一次更新可能会执行完全重新建立索引来确保一致状态。

对数据库的更改

- 插入到事件表
- 从索引日志表中删除

当使用复制捕获表时，对数据库进行了下列更改。

- 在启动初始更新之前将信号添加至 IBMSNAP_SIGNAL 表
- 在增量更新之后更改 IBMSNAP_PRUNE_SET 的同步点

HELP 命令

用途

此命令显示可用的 DB2TEXT 命令的列表，或者显示单个 DB2TEXT 命令的语法。

权限

无

命令语法



命令参数

HELP | ?
: 提供有关指定命令或原因码的帮助。

command
: 用来标识 DB2TEXT 命令的第一个关键字:

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR
- START
- STOP
- CONTROL
- ACTIVATE
- DEACTIVATE

reasoncode
: 来自 Net Search Extender 命令的原因码。

用法说明

如果除了第一个关键字之外指定了多个关键字，那么会忽略其余关键字，并且会显示所标识的命令的语法。

如果在 ? 或 HELP 后面没有指定 *command* 参数（或者根本不带参数），那么 DB2TEXT 将列示所有可用的 DB2TEXT 命令参数。

COPYRIGHT 命令

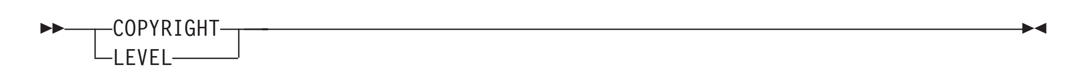
用途

提供 Net Search Extender 产品和版权信息。

权限

无

命令语法



命令参数

COPYRIGHT | LEVEL
提供产品的版本版权语句、版本号和构建信息。

Net Search Extender 在 UNIX 平台上的安装和卸载参考

本节描述 Net Search Extender 在UNIX 平台上的安装和卸载命令的语法。还包括用于显示自 V9 以来的 Net Search Extender 产品的已安装副本的 `db2nsels` 命令。

db2nse_install 命令

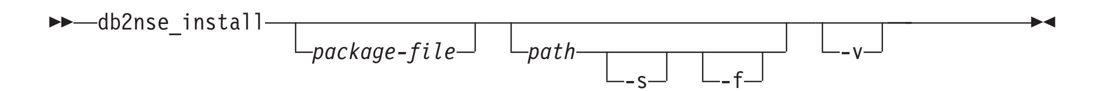
用途

此命令将在 UNIX 系统上安装新版本的 Net Search Extender。

权限

必须作为 root 用户来运行此命令。

命令语法



命令参数

package-file
包含 Net Search Extender 产品的文件的名称。

path 您现在想将 Net Search Extender 安装到的 DB2 路径。

-s 静默安装。执行检查并将日志文件写入 `/tmp` 目录。

-f 强制安装。不执行检查。

-v 显示程序版本并退出。

用法说明

此命令将安装作为一个参数传递的程序包文件中所包含的 Net Search Extender。如果不指定任何附加参数，那么安装程序将检查系统中适合于安装 Net Search Extender 的 DB2 副本。列出了所有 DB2 安装路径，但是只能选择一个可行的子集。系统将要求您选择安装路径。

如果您选择进行静默安装，那么不需要执行用户交互。将把安装过程的结果作为返回码传递给调用程序，调用程序可能是一个命令 shell 或者 shell 脚本。将由您来正确处理返回码。将把用来说明安装情况的日志文件写入 `/tmp` 目录中。

如果您选择进行强制安装，那么不需要执行用户交互，也不需要进行检查。将把安装过程的结果作为返回码传递给调用程序，调用程序可能是一个命令 shell 或者 shell 脚本。将由您来正确处理返回码。将把用来说明安装情况的日志文件写入 `/tmp` 目录中。

-s 和 -f 参数都要求传递一个 path。这些参数都没有缺省路径。

如果传递了 -v 参数，那么将只显示程序的版本，并且程序在不执行任何进一步操作的情况下就会退出。

db2nse_deinstall 命令

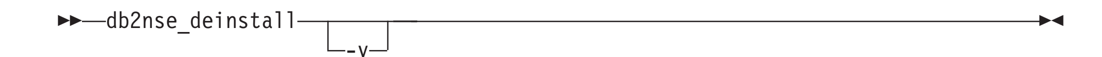
用途

此命令将除去 UNIX 系统上的 Net Search Extender。可在 DB2 副本的安装了 Net Search Extender 的每个安装子目录中执行 db2nse_deinstall 命令。

权限

必须作为 root 用户来运行此命令。

命令语法



命令参数

-v 显示程序版本并退出。

用法说明

此命令将除去自 DB2 V9 以来的 Net Search Extender。此命令将除去发出该命令的目录中的 Net Search Extender 副本。它不会除去安装在其他地方的 Net Search Extender 副本。

db2nsels 命令

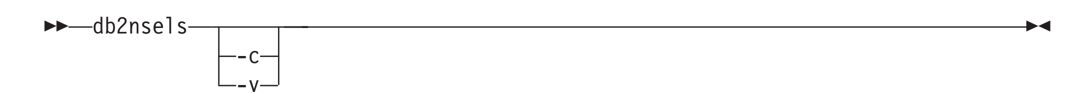
用途

此命令将显示一个表，该表中包含已安装的所有 Net Search Extender 副本。它只显示自 DB2 V9 以来的版本。db2nsels 命令位于 `/usr/local/bin` 目录中。

权限

必须作为 root 用户来运行此命令。

命令语法



命令参数

-c 在单个压缩列表中显示已安装的所有 Net Search Extender 版本（各个版本之间用冒号分隔），然后就会退出。

-v 显示程序版本（例如，9.0.0.0），然后就会退出。

用法说明

此命令将列示已安装的自 DB2 V9 以来的所有 Net Search Extender 副本。此命令的结果将显示在一个表中，或者显示在一个压缩列表中（各个项之间用冒号分隔）。这些结果包括安装的 Net Search Extender 的路径、版本和修订包编号。

例如，如果调用 db2nsels 时不带任何参数，那么输出可能如下所示:

```
db2nsels

安装路径                              级别        修订包
-----------------------------------------------------------------
/opt/ibm/db2/V9.0                  9.0.0.0      0
/test/V9.0                         9.0.0.0      0
```

如果执行该命令时使用 -c 参数，那么返回的输出是一个压缩列表，该列表中的每个信息项之间用冒号分隔。这种类型的输出便于使用一个程序或 shell 脚本来处理。例如:

```
db2nsels –c

#PATH:VRMF:FIXPACK
/opt/ibm/db2/V9.0:9.0.0.0:0
/test/V9.0:9.0.0.0:0
```

搜索自变量的语法

搜索自变量是您在文本文档中搜索词条时指定的条件。它由搜索自变量以及一个或多个搜索词条组成。

在第 59 页的『指定 SQL 搜索自变量』中以及 Net Search Extender samples 目录中的一个称为 search 的文件中提供了搜索自变量示例。

使用搜索自变量的 SQL 标量搜索函数包括:

CONTAINS
此函数使用搜索自变量来搜索特定文本文档中的文本。如果该文档包含文本或者在搜索自变量中指定的任何关系，那么将返回 INTEGER 值 1。否则，它将返回 0。

NUMBEROFMATCHES
此函数使用搜索自变量来搜索文本文档，然后返回一个指示在每个文档中找到了多少个匹配项的 INTEGER 值。

SCORE
此函数使用搜索自变量来搜索文本文档。它为找到的每个文档返回一个值，指示与同一索引中的其他文档比较起来，搜索自变量是如何描述找到的文档的。

注: 在存储过程搜索和 SQL 表值函数的搜索自变量中使用了相同的语法。

搜索自变量语法

```
►►──┬──────────────────────────┬──┬─────────────────────────────┬──────────────────►
    └─RESULT LIMIT──number─────┘  └─EXPANSION LIMIT──number─────┘
```

```
►─┬─────────────────────────────────────────────┬──────────────────────────────►
  └─STOP SEARCH AFTER──number──┬─DOCUMENT──┬─┘
                               └─DOCUMENTS─┘

►─┬─┤ boolean-search-expression ├─┬──────────────────────────────────────────►◄
  └─┤ freetext-argument ├─────────┘
```

Boolean-search-expression:

```
├─┬─┤ search-term ├──────────────────────────────────────────────────┬──────────┤
  └─┤ boolean-search-expression ├──┤ operator-or ├──┤ search-term ├─┘
```

search-term:

```
├─┬─┤ search-factor ├──────────────────────────────────────────────────┬──────────┤
  ├─┤ search-term ├──┤ operator-and ├──┤ search-factor ├──────────────┤
  ├─┤ search-term ├──┤ operator-accum ├──┤ search-factor ├────────────┤
  └─┤ search-term ├──┤ operator-minus ├──┤ positive-search-factor ├───┘
```

Search-factor:

```
├─┬─────┬─┤ positive-search-factor ├────────────────────────────────────────────┤
  └─NOT─┘
```

Positive-search-factor:

```
├─┬─┬──────────────────────────────────────────────────────┬─┤ search-primary ├─┬─────────┤
  │ │                  ┌─,─────────────────────────────┐  │                     │
  │ │                  ▼                               │  │                     │
  │ └─┬─SECTION──┬─(───"section-name"─┬──────────────┬─┴─)┘                     │
  │   └─SECTIONS─┘                    └─WEIGHT─number─┘                          │
  └─attribute-factor──────────────────────────────────────────────────────────────┘
```

Search-primary:

```
├─┬─┤ text-literal ├─────────────────────────────┬─────────────────────────────────┤
  ├─┤ context-condition ├────────────────────────┤
  ├─┤ thesaurus-invocation ├─────────────────────┤
  ├─(──┤ boolean-search-expression ├──)──────────┤
  └─(──┤ text-literal-list ├──)──────────────────┘
```

Operator-and:

```
├──&─────────────────────────────────────────────────────────────────────────────┤
```

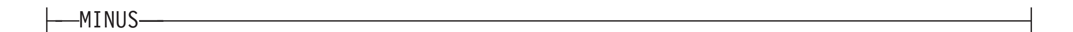
Operator-or:

```
├──|─────────────────────────────────────────────────────────────────────────────┤
```

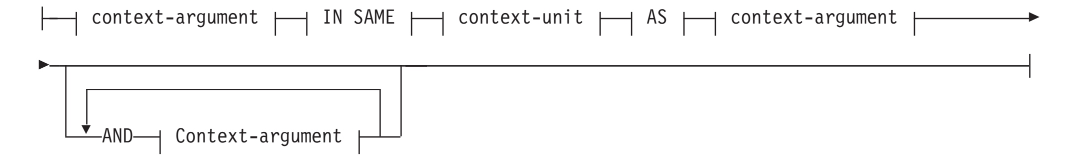
Operator-accum:

```
|--ACCUM---------------------------------------------------|
```

Operator-minus:



Context-condition:



Context-argument:

```
|----| text-literal |-------------------------------------|
  |--(--| text-literal-list |--)--|
  |--| thesaurus-invocation |----|
```

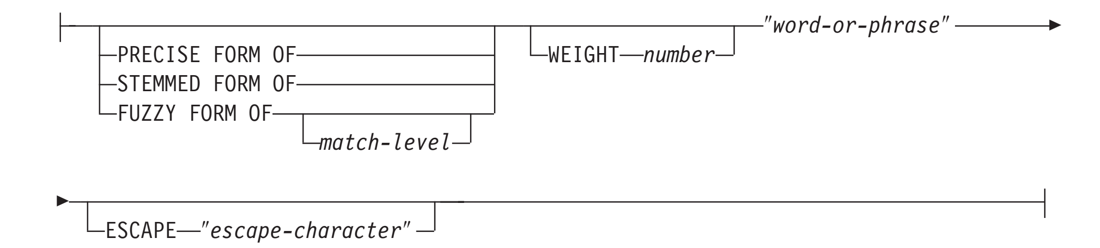
Text-literal-list:



Context-unit:

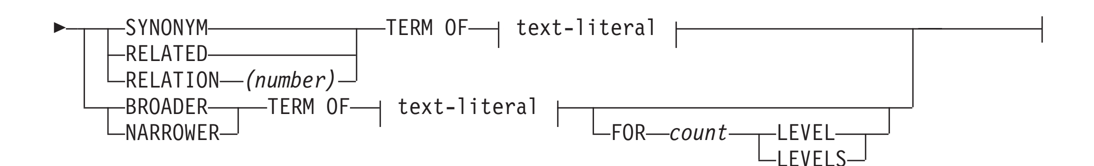
```
|----PARAGRAPH----------------------------------------------|
  |--SENTENCE--|
```

Text-literal:

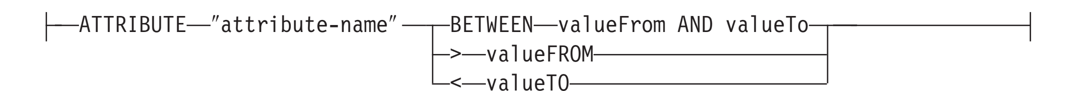


thesaurus-invocation:

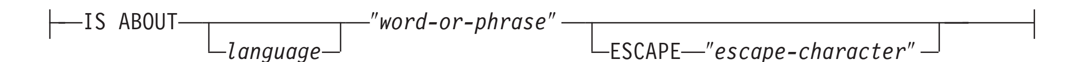
```
|--THESAURUS--"thesaurus-name"--EXPAND----------------------->
```



Attribute-factor:



freetext-argument:



示例

在第 59 页的『指定 SQL 搜索自变量』中提供了示例。

搜索自变量

参数

RESULT LIMIT *number*
: 一个关键字，它指定要由全文本搜索返回的最大结果数。

RESULT LIMIT 应当与 SCORE 函数一起使用来确保返回的结果已计算了分数，并且只处理匹配程度最高的结果。

EXPANSION LIMIT *number*
: 一个关键字，它指定可以为了进行搜索而将通配词条扩展成的最大词条数。例如，确定可以扩展搜索词条“a*”的次数。如果索引非常大，并且您要使用许多通配词条，而您又想获得更大的结果集，那么必须调整此关键字的值。扩展顺序取决于文本索引的内部组织，并且不能预先确定扩展顺序。

STOP SEARCH AFTER *number* **DOCUMENT | DOCUMENTS**
: 用来指定搜索阈值的一个关键字。如果在搜索期间达到了给定的文档数，那么搜索将停止，并返回中间结果。较小的值可提高搜索性能，但是可能会导致获得较少的结果，并且会省略可能具有高级别的文档。

注意，此关键字没有缺省值，并且 *number* 值必须是正整数。

boolean-search-expression
: 可以根据语法图并使用布尔运算符 NOT、AND、OR、ACCUM 和 MINUS 来组合搜索词条和搜索因子。这些运算符的优先顺序为下所示（第一个的优先权最高）：NOT> MINUS = ACCUM = AND > OR。可以在以下示例中了解到这种情况:

```
"Pilot" MINUS "1passenger" &"vehicle" | "transport" & "public"
```

它将按以下顺序进行求值:

```
(("Pilot" MINUS "passenger") & ("vehicle")) | ("transport" & "public")
```

如果其中一个布尔自变量求值为 true（可与 OR 运算符进行比较），那么 ACCUM 运算符求值为 true。等级值是通过累计两个操作数中的等级值来计算的。ACCUM 运算符与 AND 具有相同的绑定（优先权）。如果左操作数求值为 true，那么 MINUS 运算符求值为 true。如果右边的操作数求值为 true，那么采用左边的操作数的等级值并减去损失来计算等级值。

search-primary

如果在文档（的指定部分）中找到任何文本文字，那么由文本文字列表组成的 search-primary 求值为 true。如果在文档（的指定部分）中找到扩展的任何文本文字，那么由同义字调用组成的 search-primary 运算符求值为 true。

SECTION | SECTIONS *section-name*

一个关键字，它指定要将搜索范围限制在结构化文档中的一个或多个部分。必须在创建索引时在指定的模型文件中指定这些部分的名称，或者使用 XPath 表示法来表示这些部分。

这些部分的名称是区分大小写的。确保模型文件和查询中的部分名的大小写完全相同。

此模型描述包含可标识部分的文档的结构，因此可以单独搜索这些部分的内容。不能使用屏蔽字符来屏蔽部分名。如果在指定的其中一个部分找到了 search primary，那么使用 SECTION 子句的 *positive-search-factor* 求值为 true。

部分名并不是在执行查询期间求值的有效 XPath 表达式。如果未使用任何模型文件，那么在 XPath 表示法中将使用短语来表示缺省部分名。元素的绝对路径表达式（例如，`/father/child/grandchild`）被用作用于标识部分的名称。不支持将完整的 XPath 表达式作为部分名。

context-argument IN SAME context-unit AS context-argument AND context-argument ...

此条件允许您在同一段或同一个句子中搜索文本文字的组合。上下文自变量始终等价于文本文字列表，并且可以使用同义字扩展来将文本文字扩展为这样一个列表。

如果文档中具有上下文单元（分别为段落或句子），文档中的每个扩展的上下文自变量至少包含其中一个文本文字，那么此条件求值为 true。可以在以下示例中了解到这种情况:

```
("a","b") IN SAME PARAGRAPH AS ("c","d")
                    AND THESAURUS "t1" EXPAND SYNONYM TERM OF "e".
```

假设 e1 和 e2 是 e 的同义词，以下段落将匹配:

```
".. a c e .." ,  ".. a c e1..",  "a c e2..",
".. a d e .." ,  ".. a d e1..",  "a d e2..",
".. b c e .." ,  ".. b c e1..",  "b c e2..",
".. b d e .." ,  ".. b d e1..",  "b d e2..".
```

PRECISE FORM OF

一个关键字，使用该关键字时将完全按照输入的内容来搜索 PRECISE FORM OF 后面的单词（或者短语中的每个单词）。这种搜索是区分大小写的；即，使用的是大写字母还是小写字母非常重要。例如，如果搜索 `mice`，那么找不到“Mouse”。

此参数要求将索引配置参数 Respect case 设置为 yes。在构建索引之后就不能更改此配置设置。

STEMMED FORM OF

一个关键字，使用该关键字时会在执行搜索之前将 STEMMED FORM OF 后面的单词（或者短语中的每个单词）缩短为词干。这种搜索不区分大小写。例如，如果搜索 `mouse`，那么可以找到“Mouse”。

采用哪种方式将单词缩短其词干形式是与语言有关的。目前，只支持“英语”词干，而且单词还必须遵循常规词尾变化。

FUZZY FORM OF

用于进行“模糊”搜索的一个关键字，“模糊”搜索就是搜索与搜索词条具有相似拼写的那些词条。当在由“光学字符识别”（OCR）程序创建的文档中进行搜索时，模糊搜索特别有用。这样的文档通常包括拼写错误的单词。例如，单词 `economy` 可能会被 OCR 程序识别为 `econony`。注意，将只为文档中前三个字符相匹配的单词返回成功匹配。在上述示例中，`ecanomy` 就不是一个匹配项。如果搜索源中的单词包含屏蔽字符，那么不能使用模糊搜索。

match-level

1 到 100 之间的一个整数，它指定相似程度，其中 100 比 1 更相似。100 指定“精确匹配”，60 被认为是一个非常“模糊值”。匹配级别越模糊，耗用的搜索时间就越长，这是因为要搜索更多文档。缺省匹配级别为 70。

WEIGHT *number*

将文本文字与权值关联来更改缺省分数。允许的权值是 0（最低的分数权值）到 1000（最高的分数权值）之间的整数，缺省值为 100。

word-or-phrase

要搜索的单词或短语。在单词中可以使用的字符与语言有关。是否需要用分隔符将单词隔开，也是与语言有关的。对于英语和其他大多数语言，短语中的每个单词都必须用空格字符隔开。

要搜索包含双引号的字符串，应输入双引号两次。例如，要搜索文本 ″wildcard″ character，使用:

```
"""wildcard"" character"
```

注意，在该示例中，只能搜索一组引号。不能够搜索一个序列中的两个引号。每个单词或短语的最大长度只能为 128 个字节。

Masking characters

一个单词可以包含下列屏蔽字符:

_（下划线）

表示任何单个字符。

%（百分号）

表示任何数目的任意字符。如果一个单词只由一个 % 组成，那么它表示一个任意长度的可选单词。除了使用单个 % 来表示可选单词之外，一个单词不能只由屏蔽字符组成。如果使用了屏蔽字符，那么不能使用 THESAURUS 关键字。屏蔽字符不能跟在非字母数字字符后面。不能在模糊搜索中使用屏蔽字符。掩码始终都将只展开为单个单词。

ESCAPE *escape-character*

一个字符，它将下一个字符标识为要搜索的字符，而不是要用作屏蔽字符。例如，如果转义字符是 $，那么 $%、$_ 和 $$ 分别表示 %、_ 和 $。前面没有 $ 的任何 % 和 _ 字符都表示屏蔽字符。

在搜索期间，只允许您使用单字节转义字符。不允许使用双字节字符。

THESAURUS *thesaurus-name*
: 一个关键字，用来指定要用来扩展文本文字的同义字的名称。同义字名称是已经使用同义字编译器进行编译的同义字的文件名（不带扩展名）。它必须位于 `<os-dependent>/sqllib/db2ext/thes` 中。或者，可以在文件名前面指定完整路径。

EXPAND *relation*
: 指定使用哪种关系来扩展使用同义字的文本文字。同义字已经预定义了 DB2EXTTH 命令中描述的关系。使用下列关键字来表示这些关系:

- SYNONYM，表示等价的对称关系。
- RELATED，表示关联的对称关系。
- BROADER，可以后跟指定的深度级别的直接分层关系。
- NARROWER，可以后跟指定的深度级别的直接分层关系。

对于用户定义的关系，使用 `RELATION(number)`，它对应于 `DB2TEXTTH` 中的关系定义。

TERM OF *text-literal*

要将其他搜索词条从同义字添加至的文本文字。

count **LEVELS**

一个关键字，用来指定同义字中词条的层数（深度），这些同义字将用来扩展给定关系的搜索词条。如果不指定此关键字，那么假定 COUNT 为 1。深度的值必须是一个正整数值。

ATTRIBUTE *attribute-name*
: 搜索具有与指定条件相匹配的属性的文档。属性名是指 `CREATE INDEX` 命令中的属性表达式的名称，或者是文档模型文件中的属性定义。

属性因子仅用于类型为 double 的属性。保证该值的精度为 15 位。对于 16 位和 16 位以上的数字会进行四舍五入。在 attribute-name、valueFrom 和 valueTo 中不允许使用屏蔽字符。有关解释，请参阅下列内容:

BETWEEN *valueFrom* **AND** *valueTo*
: 如果属性的值大于（不等于） *valueFrom* 并且小于（不等于）*valueTo*，那么 BETWEEN 属性因子求值为 true。

>*valueFrom*
: 如果属性的值大于（不等于）valueFrom，那么“>”属性因子求值为 true。

<*valueTo*
: 如果属性的值小于（不等于）valueTo，那么“<”属性因子求值为 true。

如果 CREATE INDEX 命令中的属性名是使用引号来指定的，或者是在模型文件中定义的，那么指定的属性名必须精确匹配。但是，如果 `CREATE INDEX` 命令中没有指定引号，那么属性名必须采用大写。

IS ABOUT *language word-or-phrase*
: 一个允许您指定自由文本搜索自变量的选项。通过使用 IS ABOUT，就可以在文档中按任意顺序搜索您在 word-or-phrase 中指定的任何单词（但不一定是所有单词）。word-or-phrase 中使用的词条互相越接近，文档中包括的词条就更多，返回的文档的分数值就越高。

language 参数是可选的；只有对于泰国语（TH_TH）才必须设置该参数，主要是为了实现标记化。该参数对于任何其他语言都没有影响。

注意，仅当请求了文档分数值，并且搜索结果按分数值排序时，IS ABOUT 才有用。

SQL 标量搜索函数和 SQL 表值函数

Net Search Extender 提供了 SQL 标量搜索函数和 SQL 表值函数来搜索存储在 DB2 中的文本文档。

本节描述了下列 SQL 搜索函数。

搜索函数	用途
『CONTAINS 标量函数』	在特定文档中搜索文本。
第 143 页的『NUMBEROFMATCHES 标量函数』	搜索并返回找到的匹配项数目。
第 143 页的『SCORE 标量函数』	搜索并返回找到的文本文档的分数值。
第 144 页的『DB2EXT.TEXTSEARCH 命令』	SQL 表值函数返回找到的主键、匹配项数目和/或分数值组成的表。
第 146 页的『DB2EXT.HIGHLIGHT』	获取有关将文档限定为搜索结果的原因的信息。

CONTAINS 标量函数

用途

CONTAINS 标量函数将搜索由 Net Search Extender 建立索引的文本文档中的文本。如果该文档包含文本或者在搜索自变量中指定的任何关系，那么将返回 INTEGER 值 1。否则，它将返回 0。

函数语法

```
▶▶—CONTAINS—(—column-name—,—search-argument—)————————————▶◀
```

函数参数

column name

表列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument

包含要搜索的词条的类型为 VARCHAR 的字符串。

注： 不能对为视图创建的文本索引使用 CONTAINS 查询。

NUMBEROFMATCHES 标量函数

用途

NUMBEROFMATCHES 标量函数搜索文本文档，然后返回一个指示在每个文档中找到了多少个匹配项的整数值。

函数语法

```
▶▶—NUMBEROFMATCHES—(—column-name—,—search-argument—)——————————————▶◀
```

函数参数

column name
: 表列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument
: 包含要搜索的词条的类型为 VARCHAR 的字符串。

注：不能对为视图创建的文本索引使用 NUMBEROFMATCHES 查询。

SCORE 标量函数

用途

SCORE 标量函数对文本文档进行搜索，并为找到的每个文档返回一个分数值，该值指示搜索自变量是如何描述找到的文档的。

SCORE 将返回 DOUBLE 值。当搜索词条在文档中出现得更频繁时，文档的分数就会增大。

函数语法

```
▶▶—SCORE—(—column-name—,—search-argument—)——————————————▶◀
```

函数参数

column name
: 列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument
: 包含要搜索的词条的类型为 VARCHAR 的字符串。

注：不能对为视图创建的文本索引使用 SCORE 查询。SCORE 返回的值只有在与从同一索引中检索的其他值进行比较时才有意义。

DB2EXT.TEXTSEARCH 命令

用途

除了存储过程搜索和 SQL 标量搜索函数之外，Net Search Extender 还提供了与存储过程看起来非常相似的两个 SQL 表值函数。

这两个表值函数都称为 db2ext.textsearch。它们之间唯一的区别就是：一个表值函数支持 HIGHLIGHT 函数，并且具有两个附加参数：numberOfHits 和 hitInformation。

这两个表值函数都从按照以下排序标准进行排序的用户表中返回结果：在 CREATE INDEX 命令中的 INITIAL SEARCH RESULT ORDER 参数中定义的排序标准。如果包含表值函数的 SQL 查询语句在语句末尾有一个连接（例如，诸如 where T.primkey = S.key 的内容），那么结果行的顺序取决于连接方法而不是取决于在 CREATE INDEX 命令中定义的顺序。

注意，不能对具有复合主键的表使用表值函数。

> **注**：仅当用户表存储在具有单个节点的表空间中时，才能在分布式 DB2 环境中使用表值函数。还必须确保使用 DB2NODE 环境变量连接至正确的节点。

1. db2ext.textsearch without highlight support

```
        db2ext.textSearch
        (
                        query             VARCHAR(4096),
                        indexSchema       VARCHAR(128),
                        indexName         VARCHAR(128),
                        resultFirstRow    INTEGER,
                        resultNumberRows  INTEGER,
        primKeyBinding    <supported types>,// same type as primary key
        )

                        return table
        (
        primKey           <supported types>,// same type as primary key
                        numberOfMatches   INTEGER,
                        score             DOUBLE,
                        totalNbResults    INTEGER
        )
```

2. db2ext.textsearch with highlight support

```
                        db2ext.textSearch
        (
                        query             VARCHAR(4096),
                        indexSchema       VARCHAR(128),
                        indexName         VARCHAR(128),
                        resultFirstRow    INTEGER,
                        resultNumberRows  INTEGER,
        primKeyBinding    <supported types>,// same type as primary key
        numberOfHits      INTEGER
        )

                        return table
        (
        primKey           <supported types>,// same type as primary key
                        numberOfMatches   INTEGER,
```

```
                    score            DOUBLE,
                    totalNbResults   INTEGER
    hitInformation    BLOB(20K)
    )
```

函数参数

下面是一些输入参数。

query 要获取更多信息，请参阅第 135 页的『搜索自变量的语法』。

indexSchema，indexName
: 标识要搜索的索引。

resultFirstRow
: 各个部分返回的查询结果列表。此参数描述查询结果列表的哪一行是要输入到表值函数的结果表中的第一行。值必须大于等于 0。

  注意，数字 0 标识查询结果列表中的第一行。

resultNumberRows
: 此参数描述将查询结果列表中的多少行输入到了表值函数的结果表中，如果该值为 0，那么意味着所有结果都需要返回。

  注意：此参数与用来确定查询结果列表的最大大小的结果限制查询参数是不同的。

primaryKeyBinding
: 此参数的类型确定 primaryKey Output 参数的类型。如果为具有 <type1> 类型的主键的基本表创建了文本索引，那么 primaryKeyBinding 也必须是 <type1> 类型。

  另外，该参数还确定文本搜索的范围。如果 primaryKeyBinding 设置为 NULL ("CAST(NULL as <type1>)"， 那么搜索范围将是存储在索引中的所有文档。或者，可以限制只对 primaryKeyBinding 绑定至的文档进行搜索。

  例如，如果将 primaryKeyBinding 设置为 CAST(5 as BIGINT)，那么限制只对 BIGINT 主键值为“5”的单个文档进行搜索。

  注意，只支持下列类型的单个列主键：SMALLINT、INTEGER、BIGINT、REAL、DOUBLE、VARCHAR FOR BIT DATA、DATE、TIME 和 TIMESTAMP。

numberOfhits
: 此选项指定通过使用突出显示函数 db2ext.highlight 来突出显示的词条数。如果指定 0，那么会突出显示所有匹配项，最多可以有 1100 个匹配项。这个过程可能很耗时间。

函数参数

下列返回值存储在临时表中，如果请求进一步的结果，那么需要将该临时表连接至您的用户表。注意，仅当在 select 语句中请求了 NUMBEROFMATCHES、SCORE、TOTALNUMBEROFRESULTS 和 HITINFORMATION 时才会对这些函数求值。

primKey
: 找到的文档的主键。

numberofmatches
: NUMBEROFMATCHES 是一个 INTEGER 值，它指示每个文档产生了多少个匹配项。

score
: Score 将返回 DOUBLE 值。当搜索词条在文档中出现的频率提高时，文档的分数就会增大。

totalNumberOfResults
: 查询结果列表指示找到了多少个结果。注意，每一行都具有相同的值。

  还要注意，当在查询中将 STOP SEARCH AFTER 或 RESULT LIMIT 与 SCORE 语法一起使用时，此数值不再可靠。

hitInformation
: 由 db2ext.textsearch 返回的匹配项信息对于进行突出显示处理是必需的。目前，此输出参数中大约可以包含 1100 个匹配项。如果匹配项数目超过了此阈值，那么会忽略后续匹配项的匹配项信息。

  注意，仅当指定了 numberOfHits 时才会返回此值。

用法

借助 SQL 表值函数，可以按搜索存储过程搜索的相同方式来搜索视图。只有使用 SQL 表值函数 时才不需要共享内存，因此，索引不需要具有必须激活的高速缓存。

此函数主要供已经在存储过程搜索中使用了 SQL 查询的那些用户使用。但是，存在的限制就是只支持基本表的单列主键。

以下示例说明可以如何处理多列主键表:

```
select s.id from
db2ext.sample s, table (db2ext.textSearch(
                '"characteristics"',
                'DB2EXT',
                  'COMMANDS',
                  1,
                20,
                  cast(NULL as INTEGER))) t
where s.id = t.primkey
```

在此示例中，必须首先对具有单个唯一键的此表创建视图，然后对此视图创建索引。

有关将 SQL 表值函数与 db2ext.highlight 函数配合使用的示例，请参阅『DB2EXT.HIGHLIGHT』。

DB2EXT.HIGHLIGHT

用途

使用 db2ext.highlight 函数来获取一些信息，可以使用这些信息来说明将特定文档限定为搜索结果的原因。更准确地说，可以使用这些信息来:

- 获取匹配项
- 获取匹配项和周围的文本
- 在文档中的匹配项周围添加用户定义的突出显示标记。

注意，只能将 db2ext.highlight 函数与 db2ext.textsearch 表值函数配合使用。表值函数搜索索引，并将结果提供给 HIGHLIGHT 函数使用。

函数语法

```
►►—db2ext.highlight——————————————————————————————————————————►

►–(—document-content—,—hit-information—,—hit-processing-information—)——————►◄
```

函数参数

下面是一些输入参数:

document content CLOB(100K)

只支持采用 TEXT 或者已序列化的 XML 格式的 UTF8 文档。要突出显示存储在本地的 XML 文档，必须使用 XMLSERIALZE SQL/XML 函数将 XML 数据序列化为 CLOB。

要增大 CLOB 值，请使用第 128 页的『DB2EXTTH 命令』。

hit information BLOB(20K)

包含匹配项信息的字符串。如果指定了 numberOfHits 参数，那么此参数值是由 db2ext.textsearch 函数返回的。

hit processing information VARCHAR(1024)

此参数是选项值对的列表，它们之间用逗号“,”字符隔开，并将每个字符串字符用 ″ ″ 字符引起来。它指定应当如何对指定的文档处理突出显示。如果未指定任何选项，那么将按原样返回原始文档内容。

TAGS = (″STRING″, ″STRING″)

此选项使用户能够指定要插入到文档中的匹配项前面和后面的标记。如果省略了此选项，那么在文档中的匹配项前面和后面不会添加任何标记。

WINDOW_NUMBER = INTEGER

此选项指定突出显示函数应当返回文档的多少个部分（或窗口）。每个窗口都包含一个或多个匹配项，每个窗口中的第一个匹配项确定返回至用户的文档的部分。不能确定这些匹配项周围是否有文本。

如果省略此选项，那么采用 0 作为缺省值，并且会返回包含开始和结束标记（如果指定）的整个文档。在此情况下，忽略了 WINDOW_SIZE 选项。

WINDOW_SIZE = INTEGER

此选项指定建议的窗口大小（以字节计）。此实际大小可能会根据匹配项数目、匹配项长度以及开始和结束标记大小的不同而不同。如果省略了该选项，那么 0 为缺省值，并且将只返回周围没有文本的匹配项。

WINDOW_SEPARATOR = ″STRING″

此选项指定用来将一个窗口与相邻窗口隔开的标记。如果省略了该选项，那么“...”为缺省值。

FORMAT = ″STRING″

此选项指定文档的格式。有效值为 XML 或 TEXT。如果省略了此选项，那么采用 TEXT 作为缺省值。确保格式值与建立索引期间指定的值相同。

MODEL_NAME = ″STRING″
: 此选项指定与所指定的 XML 文档相关的模型名。注意，如果 FORMAT 为 TEXT，那么此选项将产生错误。

SECTIONS = (″section-name1″, ..., ″section-nameN″)
: 对于 XML 文档，可以限制只突出显示相关部分。例如，它们可以在模型文件中定义。要指定这些部分，用逗号将一个或多个部分名隔开。如果省略了此选项，那么对整个 XML 文档执行突出显示。注意，如果 FORMAT 为 TEXT，那么忽略此选项。

  DB2EXT.HIGHLIGHT 和 DB2EXT.TEXTSEARCH 函数中必须使用相同的节规范（″section-name1″,...,″section-nameN″）。

函数参数

下面是一些返回参数。

CLOB(200K)
: HIGHLIGHT 函数将返回包含由 HIGHLIGHT 函数修改的文档部分的 CLOB 值。

用法

以下示例说明了可以如何使用 HIGHLIGHT 函数:

```
select p.id,
          p.title,
       db2ext.highlight(p.content,
          t.hitinformation,
       'TAGS = ("<bf>", "</bf>"),
          WINDOW_NUMBER = 5,
          WINDOW_SIZE = 200,
          WINDOW_SEPARATOR = "...",
       FORMAT = "XML",
          SECTIONS = ("section1-name", "section2-name")')

FROM patent p, table (db2ext.textsearch(
          '"relational database systems"',
          'DB2EXT',
          'TI_FOR_CONTENT',
          0,
          20,
          CAST(NULL as BIGINT),
          15)) t

WHERE p.id = t.primkey
```

使用超过 100 KB 的文档将导致 SQL 查询终止并且产生 SQL 错误（SQL 1476N 和 sql 错误 -433）。为了避免发生这种情况，应使用 db2exthl 命令来增大允许的文档内容大小。

注: 特殊字符（例如“newline”）将按原样返回。

突出显示存储在本地的 XML 文档时，需要将这些 XML 文档序列化为 CLOB 之后才能将它们传递给 HIGHLIGHT 表值函数。以下示例说明了可以如何对使用 XMLSERIALZE SQL/XML 函数的存储在本地的 XML 文档使用 HIGHLIGHT 函数。以下样本的 patent 内容是作为本地 XML 存储的。注意，还指定了 FORMAT=″XML″:

```
select p.id,
              p.title,
              db2ext.highlight(XMLSERIALIZE(p.content AS CLOB(100K)),
              t.hitinformation,
       'TAGS = ("<bf>","</bf>"),
              FORMAT = "XML",
       SECTIONS = ("section1-name", "section2-name")')

FROM patent p, table (db2ext.textsearch(
        '"xml database systems"',
        'DB2EXT', 'TI_FOR_XML',
               0,
               20,
               CAST(NULL as BIGINT),
               15)) t

WHERE p.id = t.primkey
```

限制

- 只支持 XML 和平面文本文档。
- 只有 UTF8 数据库受支持。对于二进制文档，需要确保文档是采用 UTF8 编码的。
- 不支持泰国语文档。
- 如果建立索引期间与查询时使用的文档格式不匹配，那么 HIGHLIGHT 函数将返回不可预测的结果。
- 将只突出显示在文档的文本部分找到的匹配项。
- Highlight 函数只能与 db2ext.textsearch 函数配合使用。
- 字符串值不能包含 ″ 字符。

存储过程搜索函数

Net Search Extender 提供了一个存储过程搜索来返回预定义的结果表。结果表是在创建索引期间在高速缓存表部分中指定的。当需要按特定顺序返回少量结果时使用存储过程搜索。

例如，有这样一个 Web 应用程序，它返回具有最佳分数的前 20 行，然后按照每次返回 20 行的方式返回其余行。

> **注:** 仅当用户表存储在单节点表空间中时，才能在分布式 DB2 环境中使用存储过程函数。
>
> 还必须确保使用 DB2NODE 环境变量连接至正确的节点。

存储过程搜索的 DB2EXT.TEXTSEARCH

函数语法

```
db2ext.TextSearch(

                IN      query                   VARCHAR(4096),
                IN      indexSchema             VARCHAR(128),
                IN      indexName               VARCHAR(128),
                IN      resultFirstRow          INTEGER,
                IN      resultNumberRows        INTEGER,
                IN      scoringFlag             INTEGER,
                IN      searchTermCountsFlag    INTEGER,
                OUT     searchTermCounts        VARCHAR(4096),
                OUT     totalNumberOfResults    INTEGER )
```

函数参数

下面是一些输入参数。

Query 有关进一步的信息，请参阅第 135 页的『搜索自变量的语法』。

indexSchema, indexName
标识要搜索的索引。

resultFirstrow
采用多个部分来返回查询结果列表。该参数描述查询结果列表的哪一行是要放置到存储过程的结果集中的第一行。查询结果列表中的第一行是用数字 0 标识的。

resultNumberRows
此参数描述将查询结果列表中的多少行放置到存储过程的结果集中。

这不会与查询中的 ″result limit″ 表达式混淆，该表达式确定查询结果列表的最大大小。

该值应当大于等于零。其中零表示需要返回所有结果。

注: 如果请求了一个较大的结果集，那么应确保提供了临时用户表空间。如果没有提供临时表空间，那么创建一个表空间。以下示例将在 UNIX 平台上创建一个表空间:

```
db2 "create user temporary tablespace tempts managed by system
                        using ('/work/tempts.ts')"
```

scoringFlag
0 表示不计算分数，而 1 表示计算分数。如果请求了计算分数，那么会返回一个具有分数值的附加列，首先填写的是最高值。

searchTermCountsFlag
此参数将控制 searchTermCounts 处理。如果 searchTermCountsFlag 为 0， 那么不计算 searchTermCounts。

函数参数

下面是一些输出参数。

searchTermCounts
索引中每个搜索词条查询出现的次数。这些计数按照查询中搜索词条的顺序作为用空格隔开的列表来返回。

有关信息，请参阅 **searchTermCountsFlag**。

totalNumberOfResults
在查询结果列表中找到的总结果数。

还要注意，当在查询中将 STOP SEARCH AFTER 或 RESULT LIMIT 与 scoringFlag 语法一起使用时，此数值不再可靠。

用法

结果集中由存储过程返回的列是由 DB2TEXT CREATE INDEX 命令的 CACHE TABLE 选项给定的。如果 scoringFlag=1，那么会添加类型为 double 的一列。此列包含 SCORE 值。

使用下列选项来提高与第一个查询具有相同字符串的第二个查询的性能。注意，这必须是在不需要 totalNumberOfResults 的另一个游标窗口中:

- 如果不需要计算分数，那么添加以下语法：STOP SEARCH AFTER x DOCUMENTS，其中 x 是 resultFirstRow + resultNumberRows。
- 如果需要计算分数，那么添加以下语法：STOP SEARCH AFTER y DOCUMENTS，其中 y 等于第一个查询中的 totalNumberOfResults。

为了确保连接至正确的节点以进行搜索，可能必须设置 DB2NODE 环境变量。

对于 UNIX，使用以下命令:

```
export DB2NODE=<no>
```

注意，所有物理节点都具有同步时间很重要。

对于 Windows，使用以下命令:

```
set DB2NODE= <no>
```

注：如果受防护用户标识与实例所有者标识不同，那么它不可使用分区数据库。

Net Search Extender 消息

Net Search Extender 提供下列消息类型:

- 参考消息和警告消息
- 错误消息

注意，从搜索函数返回的 SQL 状态是 38600 和 CTE 错误号。

参考消息和警告消息

CTE0001
操作成功完成。

CTE0002
更新和锁定服务已启动且正在运行。

CTE0003
索引更新已开始。

CTE0004
索引更新已结束。

CTE0005
索引更新落实：成功插入、更新和/或删除了“%1”、“%2”和“%3”文档。

CTE0006
访问文本索引时出现问题。有关详细信息，检查 db2diag.log。

CTE0007
节“%1”没有出现在这些文档的任何一个中，或它是无效文档模型节名。

CTE0008
索引重组已开始。

CTE0009

索引重组已结束。

CTE0010

属性“%1”无效。

CTE0011

高速缓存激活已开始。

CTE0012

高速缓存激活已结束。

CTE0013

持久高速缓存已除去。

CTE0014

高速缓存已取消激活。

错误消息 CTE0100 - CTE0199

本节包含 Net Search Extender 生成的错误消息

- 第 159 页的『CTE0127: 发生了更新和锁定服务错误。原因码: “%1”。』
- 第 159 页的『CTE0129: 不允许将空值作为参数传递。』
- 第 159 页的『CTE0130: 指定的搜索自变量超出最大长度。当前搜索自变量长度为“%1”, 而最大受支持长度为“%2”。』
- 第 160 页的『CTE0131: 用户定义的函数“%1”.“%2”不存在。』
- 第 160 页的『CTE0132: 文本索引“%1”.“%2”不存在。』
- 第 160 页的『CTE0133: 文本索引“%1”.“%2”已存在。』
- 第 160 页的『CTE0135: 对象“%1”.“%2”不存在。』
- 第 160 页的『CTE0136: 列“%1”不存在于“%2”中。“%4”中定义“%1”。』
- 第 161 页的『CTE0137: 表空间“%1”不存在。』
- 第 161 页的『CTE0138: 表空间“%1”不是常规表空间。』
- 第 161 页的『CTE0139: 未设置环境变量“%1”。』
- 第 161 页的『CTE0140: 已对文本启用数据库“%1”。』
- 第 161 页的『CTE0141: 未对文本启用数据库“%1”。』
- 第 161 页的『CTE0142: 命令需要授予用户“%3”对“%1”.“%2”的控制权限。』
- 第 162 页的『CTE0143: 对于用户“%1”, 命令需要数据库管理权限。』
- 第 162 页的『CTE0144: 在数据库“%1”中至少有一个文本索引是活动的。』
- 第 162 页的『CTE0145: CCSID “%1”不受支持。』
- 第 162 页的『CTE0146: 语言“%1”不受支持。』
- 第 162 页的『CTE0147: 格式“%1”不受支持。』
- 第 163 页的『CTE0148: 指定的格式“%1”不接受模型文件。』
- 第 163 页的『CTE0149: 对索引更新频率指定的词条过多 (从“%1”开始)。』
- 第 163 页的『CTE0150: 命令意外结束。请检查命令语法。』
- 第 163 页的『CTE0151: 标记“%1”意外出现。请检查命令语法。』
- 第 163 页的『CTE0152: 标记“%1”过长。』
- 第 163 页的『CTE0153 : 标记“%1”在更新频率中出现两次。』
- 第 164 页的『CTE0154: “%2”的值“%1”超出范围。T有效范围是“%3”至“%4”。』
- 第 164 页的『CTE0155: 搜索字符串为空。』
- 第 164 页的『CTE0157: “%1”附近存在语法错误。』
- 第 164 页的『CTE0158 : 缺少自由文本搜索字符串。』
- 第 164 页的『CTE0159 : 搜索字符串超出允许的长度“%1”。』
- 第 164 页的『CTE0160 : 尚未在搜索字符串中指定任何节名。』
- 第 165 页的『CTE0162: 未能处理转义命令。』
- 第 165 页的『CTE0163: 在同义字子句中未指定任何同义字名称。』
- 第 165 页的『CTE0164 : 同义字关系“%1”中存在语法错误。』
- 第 165 页的『CTE0166: 自由文本一定要是搜索查询中的最后一个语句。』
- 第 165 页的『CTE0167 : 自由文本查询“%1”中存在语法错误。』
- 第 165 页的『CTE0168 : 节语句中缺少左括号。』
- 第 166 页的『CTE0169: 节语句中缺少逗号或右括号。』
- 第 166 页的『CTE0170: 缺少右双引号。』

- 第 166 页的『CTE0171: 缺少节名的左双引号。』
- 第 166 页的『CTE0172: 缺少节名的右双引号。』
- 第 166 页的『CTE0173: 必须在转义子句中定义一个转义字符。』
- 第 166 页的『CTE0174: 不允许将空白字符作为转义字符。』
- 第 167 页的『CTE0175: 定义了转义子句但在搜索短语中找不到任何屏蔽字符。』
- 第 167 页的『CTE0176：短语中的转义字符的后续字符既不是同一字符也不是屏蔽字符。』
- 第 167 页的『CTE0177: 编号值“%1”无效。』
- 第 167 页的『CTE0178: 模糊短语中的屏蔽字符前必须加上转义字符。』
- 第 167 页的『CTE0179: 同义字名称“%1”超出允许的长度“%2”。』
- 第 168 页的『CTE0180: 找不到同义字“%1”。』
- 第 168 页的『CTE0181: 不能装入库“%1”。』
- 第 168 页的『CTE0182: 不能从库“%2”装入函数“%1”。』
- 第 168 页的『CTE0183: 使用共享系统资源时发生了错误。』
- 第 168 页的『CTE0184: 未发出任何 db2text start 命令。』
- 第 168 页的『CTE0185: 更新和锁定服务已经处于活动状态。』
- 第 169 页的『CTE0186: 发生了更新和锁定服务错误，有关详细信息，请检查 db2diag.log。』
- 第 169 页的『CTE0187: 更新和锁定服务仍然是活动的，使用 FORCE 选项来停止这些服务。』
- 第 169 页的『CTE0188: 使用更新和锁定服务时出现临时问题。请重试。』
- 第 169 页的『CTE0189: 找不到可执行程序“%1”。』
- 第 169 页的『CTE0190: 不能启动可执行程序“%1”。』
- 第 170 页的『CTE0191：删除索引操作未完成。有关详细信息，请检查 db2diag.log。』
- 第 170 页的『CTE0192: 更新索引操作中发生了错误。请检查事件表“%1”.“%2”和 db2diag.log 以了解详细信息。』
- 第 170 页的『CTE0194: 列“%2”的类型“%1”不受支持。』
- 第 170 页的『CTE0195: “%1”不是绝对路径。』
- 第 170 页的『CTE0198: 没有任何相应的文本索引。』
- 第 171 页的『CTE0199: 没有与表“%2”的列“%1”相对应的文本索引。』

CTE0100: DB2 操作失败。DB2 信息：“%2”“%4”。

说明

发生了 DB2 错误，不允许进一步处理。

措施

要了解有关此 DB2 错误的更详细信息，可使用 `db2 ? SQLxxx`。

CTE0101: 搜索引擎操作失败。原因码: “%2”、“%3”、“%4”、“%5”和“%6”。

说明

发生了“搜索引擎”错误，不允许进一步处理。

措施

有关更详细的信息，请参阅“搜索引擎”原因码描述。

CTE0102: 常规系统函数失败。错误: “%2”。

说明

发生了系统错误，不允许进一步处理。

措施

可以在 UNIX 上的 `errno.h` 头文件中找到更多信息。

CTE0103: 发生了内部错误。位置: “%1”和“%2”。

说明

发生了内部处理错误，不允许进一步处理。尝试启动并停止更新和锁定服务以及 DB2。

措施

如果仍然存在错误，那么启动跟踪并检查 `db2diag.log`。

CTE0104: 内存分配错误（搜索引擎）。

说明

系统的内存不足。

措施

增加实例所有者的可用内存大小或停止并行运行的其他进程。

CTE0105: 内存分配错误。

说明

系统的内存不足。

措施

增加用户的可用内存大小或停止并行运行的其他进程。

CTE0106: 表“%1”.“%2”没有主键。

说明

已尝试对没有主键的表创建索引。

措施

调用 `db2 alter table` 以确保主键存在。然后再次尝试创建索引。

CTE0107: 目录“%1”不存在。

说明

您指定了不存在的目录。

措施

创建目录，并确保实例所有者可以对其进行访问。然后再次尝试指定目录。注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

CTE0108: 对象“%1”.“%2” 的键列的内部大小“%4”比允许的最大大小“%3”还要大。

说明

键列的内部表示法超出最大大小。

措施

再次创建索引之前更改表的布局。使用较小的键列，这还可以提高性能。

CTE0109: 对象“%1”.“%2”的键列“%3”数比允许的最大值“%4”还要大。

说明

最多支持 14 个键列。

措施

再次创建索引之前更改表的布局。

CTE0110: 对象“%1”.“%2”的主键大于允许的值。

说明

主键长度超过了最大大小。主键长度限制取决于表所使用的表空间的页大小:

```
最大键长度          页大小
 --------------   ---------
   1007             4K
   2031             8K
   4079             16K
 4094             32K
```

注意，如果主键中包含多列，那么上面所提到的长度限制应减小，每个附加列减少两个字节。

措施

再次创建索引之前更改表的布局。使用较小的键列，这还可以提高性能。

CTE0111: 文件“%1”不可读。

说明

不能读取指定的文件。

措施

检查文件的访问权。考虑将“存储过程”作为受防护的用户标识来运行，这可能还需要使用此文件的权限。

CTE0112: 不能打开文件“%1”。

说明

打不开指定的文件。

措施

验证是否正确地指定了文件。

CTE0113: 将模型文件“%1”转换为 UTF8 编码时出错。

说明

指定的 CCSID 或缺省数据库 CCSID 与模型文件 CCSID 不匹配。

措施

确保正确指定模型文件 CCSID。

CTE0114: 无法在文件“%2”中注册文档模型“%1”。

说明

未能使用模型文件。

措施

检查是否正确地指定了模型文件语法。

CTE0115: 发生了锁定问题。锁管理器信息：“%1”“%2”。

说明

发生了内部锁定问题。

措施

使用 `db2text control` 命令来检查当前锁定。使用同一命令，清除暂挂锁定。如果这样作没什么用，那么停止并重新启动锁定和更新服务。

CTE0116: 操作与现有锁定冲突。

说明

已尝试在对索引运行其他命令时使用当前不允许的命令。

措施

检查对此索引持有的锁定以了解当前正在运行哪些命令。等待直到其他命令完成。如果操作不再运行而锁定仍然活动，那么清除索引的锁定并重试。

CTE0117: 已使用数据库的所有可用锁定空间。请更改配置。

说明

已尝试使用超过锁文件中配置的数目的数据库。

措施

在锁定配置 db2extlm.cfg 中更改想要并行使用的数据库的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0118: 已使用数据库的索引的所有可用锁定空间。请更改配置。

说明

已尝试对一个数据库使用超过在锁文件中配置的数目的索引。

措施

在锁定配置文件 db2extlm.cfg 中更改想要并行使用的索引的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0119: 已使用索引的锁定的所有可用空间。

说明

正在运行的操作针对一个索引需要的锁定比锁定配置文件中配置的还要多。

措施

在锁定配置 db2extlm.cfg 中更改想要并行使用的锁定的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0120: 更新和锁定服务配置文件错误。

说明

配置文件 db2extlm.cfg 出错。

措施

检查 db2extlm.cfg 文件并更正错误。使用 db2text start 命令重新启动更新和锁定服务。

CTE0121: 打不开更新和锁定服务配置文件。

说明

打不开 db2extlm.cfg 文件。

措施

检查文件是否存在以及是否可访问。如果不能访问文件，尝试使用 `db2iupdt` 更新 db2 实例。

CTE0122: 在更新和锁定服务配置文件中发现语法错误。

说明

在更新和锁定服务配置文件中发现语法错误。

措施

检查更新和锁定服务配置文件以查找错误。

CTE0126: 更新和锁定服务输入文件“%1”已毁坏。

说明

更新和锁定服务的必需文件已毁坏。

措施

检查文件是否存在以及是否可访问。如果可访问该文件，那么重命名该文件并重新启动更新和锁定服务。应该再次创建该文件。但是，此操作将除去创建索引的所有指定频率更新。

CTE0127: 发生了更新和锁定服务错误。原因码：“%1”。

说明

在更新和锁定服务区域中发生了内部错误。

措施

停止 DB2 和 Net Search Extender，然后清除共享资源。再次尝试启动这两项。如果这样做不起作用，那么向 IBM® 代表报告该问题。

CTE0129: 不允许将空值作为参数传递。

说明

DB2 将空值传递至用户定义的内部函数。

措施

首先确保指定基本表具有主键。更改选择语句以避免此问题。打开跟踪功能，并将返回的信息传递给 IBM 服务中心。

CTE0130: 指定的搜索自变量超出最大长度。当前搜索自变量长度为“%1”，而最大受支持长度为“%2”。

说明

指定的搜索自变量的长度为“%1”。最大长度不能超过“%2”。

措施

将搜索自变量的长度减少至“%2”。

CTE0131: 用户定义的函数“%1”.“%2”不存在。

说明

指定的用户定义的函数在此数据库中不存在。

措施

检查为此用户定义的函数指定的名称，或在正使用的数据库中注册该用户定义的函数。

CTE0132: 文本索引“%1”.“%2”不存在。

说明

指定的文本索引在此数据库中不存在。

措施

检查指定的名称和正在使用的数据库。使用 db2ext.textcolumns 视图来查看现有文本索引。

CTE0133: 文本索引“%1”.“%2”已存在。

说明

指定的文本索引在此数据库中已存在。

措施

检查指定的名称和正在使用的数据库。使用 db2ext.textcolumns 视图来查看现有文本索引。

CTE0135: 对象“%1”.“%2”不存在。

说明

指定的对象名不存在于此数据库中。

措施

检查指定的对象名和正在使用的数据库。

CTE0136: 列“%1”不存在于“%2”中。“%4”中定义“%1”。

说明

指定的列不存在。

措施

检查指定的列名。检查正在使用的表、视图或数据库。

CTE0137: 表空间“%1”不存在。

说明

指定的表空间不存在于此数据库中。

措施

检查指定的名称和正在使用的数据库。

CTE0138: 表空间“%1”不是常规表空间。

说明

指定的表空间不是常规表空间。只能在常规表空间中创建事件表。

措施

再次将此命令与常规表空间配合使用。

CTE0139: 未设置环境变量“%1”。

说明

未设置必需的环境变量。

措施

检查环境，指定必需的变量，并再次使用该命令。

CTE0140: 已对文本启用数据库“%1”。

说明

已对文本启用指定数据库。

措施

检查指定的名称。还要检查隐含隐式连接的 DB2DBDFT 变量。

CTE0141: 未对文本启用数据库“%1”。

说明

未对文本启用指定数据库。

措施

检查指定的数据库名称和 DB2DBDFT 变量。如果数据库名称正确，那么使用命令 db2text enable database for text。

CTE0142: 命令需要授予用户“%3”对“%1”.“%2”的控制权限。

说明

您没有使用此命令的权限。

措施

只有此表的所有者才能使用此命令或为您提供必需的权限。

CTE0143: 对于用户“%1”，命令需要数据库管理权限。

说明

您没有使用此命令的必需权限。

措施

只有数据库的所有者才能使用此命令或为您提供必需的权限。

CTE0144: 在数据库“%1”中至少有一个文本索引是活动的。

说明

在删除所有文本索引之前，不能禁用数据库。

措施

请参阅 `db2ext.textcolumns` 视图以获取现有索引。使用 DROP INDEX 命令删除现有索引或使用 DISABLE DATABASE 命令指定 FORCE 选项。

CTE0145: CCSID “%1”不受支持。

说明

指定的 CCSID 不受支持。

措施

指定有效的 CCSID。

CTE0146: 语言“%1”不受支持。

说明

指定的语言不受支持。

措施

指定有效的语言。

CTE0147: 格式“%1”不受支持。

说明

指定的格式不受支持。

措施

指定有效的格式。

CTE0148: 指定的格式“%1”不接受模型文件。

说明

格式“%1”不支持模型文件。

措施

使用接受模型文件的格式或从命令中除去模型文件。

CTE0149: 对索引更新频率指定的词条过多（从“%1”开始）。

说明

更新频率的语法不正确。

措施

确保只指定 DAY、HOUR 和 MINUTE 参数一次。

CTE0150: 命令意外结束。请检查命令语法。

说明

命令语法不正确。

措施

检查命令语法。验证是否指定了必需的参数。

CTE0151: 标记“%1”意外出现。请检查命令语法。

说明

命令的语法不正确。

措施

检查命令语法，并验证正在使用的标记在特定命令中是否允许使用。

CTE0152: 标记“%1”过长。

说明

标记过长。

措施

检查命令语法，并验证标记是否减少到允许的最大大小。

CTE0153 : 标记“%1”在更新频率中出现两次。

说明

对更新频率指定了不正确的语法。

措施

确保只指定 DAY、HOUR 和 MINUTE 参数一次。

CTE0154: “%2”的值“%1”超出范围。T有效范围是“%3”至“%4”。

说明

指定了不正确的值。值应该在允许的范围内。

措施

更新命令。更改该值以使它与允许范围内的值相匹配。

CTE0155: 搜索字符串为空。

说明

指定了空的搜索字符串。

措施

检查搜索字符串是否包括有效的字母数字字符。

CTE0157: “%1”附近存在语法错误。

说明

指定了不正确的搜索语法。

措施

检查 %1 附近的语法。更正错误并重试。

CTE0158 : 缺少自由文本搜索字符串。

说明

指定自由文本字符串。

措施

检查“is about”后面的搜索字符串是否包括有效字母数字字符。

CTE0159 : 搜索字符串超出允许的长度“%1”。

说明

搜索字符串过长。

措施

减少搜索字符串的大小并重试。

CTE0160 : 尚未在搜索字符串中指定任何节名。

说明

需要指定有效节名。

措施

添加有效节名并重试。

CTE0162: 未能处理转义命令。

说明

搜索字符串包括太多可用作屏蔽字符的特殊字符。

措施

减少搜索词条中的特殊字符数，或避免使用转义命令。可以使用下列特殊字符：! * + , _ . : ; { } ~ | ? [ ] ` = \

CTE0163: 在同义字子句中未指定任何同义字名称。

说明

请求了同义字搜索而未使用同义字名称。

措施

在搜索自变量中指定同义字名称。

CTE0164 : 同义字关系“%1”中存在语法错误。

说明

同义字关系的指定语法不正确。

措施

按照语法规范更新同义字关系。

CTE0166: 自由文本一定要是搜索查询中的最后一个语句。

说明

“is about”标记后面不允许再有运算符。

措施

重写查询字符串。最后一个运算符一定要是“is about”。

CTE0167 : 自由文本查询“%1”中存在语法错误。

说明

自由文本字符串的语法不正确。

措施

按照语法规范更新自由文本字符串。

CTE0168 : 节语句中缺少左括号。

说明

节语句的语法不正确。

措施

按照语法规范更新节语句。

CTE0169: 节语句中缺少逗号或右括号。

说明

节语句的语法不正确。

措施

按照语法规范更新节语句。

CTE0170: 缺少右双引号。

说明

搜索词条的指定语法不正确。

措施

按照语法规范更新搜索词条。

CTE0171: 缺少节名的左双引号。

说明

节语句的语法不正确。

措施

按照语法规范更新节语句。

CTE0172: 缺少节名的右双引号。

说明

节语句的语法不正确。

措施

按照语法规范更新节语句。

CTE0173: 必须在转义子句中定义一个转义字符。

说明

转义子句中只能存在一个字符。

措施

除去转义子句中的附加字符。

CTE0174: 不允许将空白字符作为转义字符。

说明

在转义子句中不允许有空白字符。

措施

将转义子句更改为具有有效字符的子句。

CTE0175: 定义了转义子句但在搜索短语中找不到任何屏蔽字符。

说明

在未使用屏蔽字符的情况下指定了转义子句。

措施

除去转义子句。

CTE0176 : 短语中的转义字符的后续字符既不是同一字符也不是屏蔽字符。

说明

转义字符后面的字符必须是屏蔽字符或转义字符本身。

措施

更改搜索字符串以正确使用转义字符。

CTE0177: 编号值“%1”无效。

说明

搜索自变量中的指定编号无效。

措施

检查文档以了解有效范围。更新搜索自变量中的值。

CTE0178: 模糊短语中的屏蔽字符前必须加上转义字符。

说明

不允许屏蔽与模糊搜索一起使用。

措施

使用转义字符更新搜索字符串。

CTE0179: 同义字名称“%1”超出允许的长度“%2”。

说明

不支持长度超过 60 个字节的主键。

措施

再次创建索引之前更改表的布局。

CTE0180: 找不到同义字“%1”。

说明

找不到指定的同义字。

措施

检查同义字文件是否位于同义字目录中，或者它是否是标准的。

CTE0181: 不能装入库“%1”。

说明

找不到库。

措施

检查库是否位于库路径中以及是否可用。启动并停止 DB2 以确保使用当前设置。

CTE0182: 不能从库“%2”装入函数“%1”。

说明

不能装入库入口点。

措施

访问的库可能是无效的。检查该库是否只指定了一次。

CTE0183: 使用共享系统资源时发生了错误。

说明

不能实现对共享系统资源（例如，共享内存或信号量）的请求。

措施

检查当前系统状态和配置。在 UNIX 上，使用 `ipcs` 命令来检查资源。停止所有应用程序（例如，DB2 和 Net Search Extender）。如果列示了更多资源，那么使用 `ipcrm` 来清除。

CTE0184: 未发出任何 db2text start 命令。

说明

调用了需要锁定和更新服务的命令。

措施

使用 `db2text start` 启动更新和锁定服务。

CTE0185: 更新和锁定服务已经处于活动状态。

说明

发出了 `db2text start`，但是更新和锁定服务已在运行。

措施

没有任何进一步的操作。

CTE0186: 发生了更新和锁定服务错误，有关详细信息，请检查 db2diag.log。

说明

发生了更新和锁定服务错误。

措施

检查 `db2diag.log` 以了解进一步的信息，或者清除共享资源。另请参阅 CTE0183。

CTE0187: 更新和锁定服务仍然是活动的，使用 FORCE 选项来停止这些服务。

说明

`db2text stop` 命令尚未停止锁定服务，仍然有进程正在运行。

措施

使用 `db2text control` 检查哪些进程正在运行并等待这些进程完成。如果需要停止它们，那么使用 FORCE 选项。

CTE0188: 使用更新和锁定服务时出现临时问题。请重试。

说明

`db2text stop` 命令尚未停止锁定服务。程序仍然在运行或发现不一致状态。

措施

使用 `db2text control` 检查哪些进程正在运行并等待这些进程完成。要停止它们，使用 FORCE 选项。

CTE0189: 找不到可执行程序“%1”。

说明

找不到程序文件或不能访问该文件。

措施

检查程序文件是否位于 DB2 服务器的 `bin` 或 `adm` 目录中。如果找不到该文件，那么安装会毁坏。

CTE0190: 不能启动可执行程序“%1”。

说明

不能启动程序。

措施

检查程序是否位于 DB2 服务器的 bin 或 adm 目录中并且安装了适当的库。有关进一步信息，在服务器上手动调用该程序。

CTE0191 ：删除索引操作未完成。有关详细信息，请检查 db2diag.log。

说明

删除索引操作未完成，可能是由 FORCE 选项导致的。

措施

使用 FORCE 选项会删除所有内容而不考虑任何错误。检查索引目录以找出暂挂文件并手动除去这些文件。

CTE0192: 更新索引操作中发生了错误。请检查事件表“%1”.“%2”和 db2diag.log 以了解详细信息。

说明

在索引更新过程期间，所有文档错误都会写至事件表。

措施

有关文档错误的更多信息，检查事件表。在修正问题之后清除事件日志。

CTE0194: 列“%2”的类型“%1”不受支持。

说明

使用的列不在受支持列的列表中。

措施

检查创建索引以获取“键和建立索引”的有效列的列表。对命令作适当更改并重试。

CTE0195: “%1”不是绝对路径。

说明

服务器上的绝对路径是必需的。

措施

检查路径并在命令中填写绝对路径。

CTE0198: 没有任何相应的文本索引。

说明

列没有任何文本索引。

措施

检查文本索引是否仍然存在。

CTE0199: 没有与表“%2”的列“%1”相对应的文本索引。

说明

已尝试搜索没有文本索引的列。

措施

检查正在搜索的列或对该列创建文本索引。

错误消息 CTE0200 - CTE0360

本节包含 Net Search Extender 生成的错误消息

- 第 179 页的『CTE0231: 未在表空间“%2”.“%3”所在的节点组“%4”中定义“%1”。』
- 第 179 页的『CTE0232: 所指定或缺省表空间“%1”不是单节点表空间。这对于视图的索引或在指定 CACHE TABLE 选项时是必需的。』
- 第 179 页的『CTE0233: 正在运行的管理命令存在冲突。请稍后重试此命令。』
- 第 180 页的『CTE0234: 正在对文本索引运行的管理命令存在冲突。请稍后重试此命令或指定 DISABLE DATABASE 命令的 FORCE 选项。』
- 第 180 页的『CTE0235: 找不到 Net Search Extender 的任何有效许可证。』
- 第 180 页的『CTE0236: 对于 MPP 实例，只有 Node0 是受支持的。』
- 第 180 页的『CTE0237: 内部错误: 日志表“%1”.“%2”包含无效操作“%3”。』
- 第 180 页的『CTE0238: 内部错误: 表“%1”.“%2”的列“%3”中包含不正确的语法表达式。』
- 第 181 页的『CTE0239: 内部错误: 索引属性“%1”的总长度超出最大长度“%2”。』
- 第 181 页的『CTE0240: 内部错误: 设置环境变量“%1”失败。』
- 第 181 页的『CTE0242: 参数“%2”的值“%1”无效。』
- 第 181 页的『CTE0243: 文本索引“%1”.“%2”尚未被激活。』
- 第 181 页的『CTE0244: 内部错误: 对“%1”的调用返回 rc=″%2″，SQLCODE=″%3″。』
- 第 182 页的『CTE0245: 请求的高速缓存大小超出可用的高速缓存大小。将最大高速缓存大小增加至大于“%1”的值或降低 pctfree 值。』
- 第 182 页的『CTE0246: 文件“%1”为空。』
- 第 182 页的『CTE0247: 未能创建 Net Search Extender 存储过程。』
- 第 182 页的『CTE0248: 生成的搜索字符串过长。请降低搜索查询的复杂性。』
- 第 182 页的『CTE0249: 可执行程序“%1”异常终止。』
- 第 183 页的『CTE0250 : 列类型变换函数“%2”.“%3”的返回类型“%1”不受支持。』
- 第 183 页的『CTE0251: 内部错误: 列类型“%1”不受支持。』
- 第 183 页的『CTE0252: 缺少参数“%1”。』
- 第 183 页的『CTE0253: 找不到日志视图中列示的文档。』
- 第 183 页的『CTE0254: 索引“%1”的高速缓存已激活。』
- 第 184 页的『CTE0255 : 缺少高速缓存结果列表达式的列名。将“ AS <cache column name>”添加至表达式。』
- 第 184 页的『CTE0256: 为建立索引选择数据所需的查询失败。请降低属性、高速缓存表或初始搜索结果顺序表达式的复杂性。』
- 第 184 页的『CTE0257: 创建共享内存时出错。』
- 第 184 页的『CTE0258: 共享内存版本错误。』
- 第 184 页的『CTE0259: 不能在全局共享内存中插入条目。条目已存在。』
- 第 185 页的『CTE0260: 不能访问全局共享内存中的条目。找不到条目。』
- 第 185 页的『CTE0261: 对于此实例中的文本索引，至少存在一个激活的高速缓存。使用 DEACTIVATE CACHE 命令取消激活任何已激活的索引的高速缓存，或者使用 FORCE 选项来停止。』
- 第 185 页的『CTE0262: 参数“%1”的值过长。』

- 第 191 页的『CTE0295: 为非二进制文本列指定了无效 CCSID“%1”。』
- 第 192 页的『CTE0296: 在“%2”上找不到库“%1”。请检查 Net Search Extender 安装。』
- 第 192 页的『CTE0360: 显示了特定错误消息』

CTE0200: 必须至少指定一个命令选项。

说明

ALTER INDEX 命令更改索引的特征（例如，更新和存储器选项）。尚未指定要更改的任何特征。

措施

指定至少一个命令选项。有关所有可能的选项，请参阅命令语法。

CTE0201: 存在与同一列上的现有文本索引的冲突。

说明

对同一列定义的文本索引是使用此创建索引命令中的不同参数创建的。

措施

在创建索引命令中更正参数值。确保下列参数对现有索引和要创建的索引具有相同值：ccsid、语言、格式、文档模型、索引配置、列函数和属性。

CTE0202: 指定键列时对象“%1”.“%2”必须是视图。

说明

指定的对象不是视图。对视图的列建立索引时，只允许使用 KEY COLUMNS FOR INDEX ON VIEW 子句。

措施

除去 KEY COLUMNS FOR INDEX ON VIEW(SQL-columname-list) 子句。

CTE0203：未使用 CACHE TABLE 选项创建文本索引“%1”.“%2”。执行命令时需要使用此选项。

说明

只有使用 CACHE TABLE 选项创建指定的索引，才能执行此命令。

措施

使用 CACHE TABLE 选项创建索引。有关命令语法，请参阅文档。

CTE0204: 缺少属性名。将“AS <attribute name>”添加至属性表达式。

说明

每当在属性表达式中使用列表达式时，都必须提供属性名。例如：（C1+C2 AS myname）。

措施

将“AS <attribute name>”添加至属性表达式。

CTE0205: CACHE TABLE 表达式无效。

说明

高速缓存表表达式中的列列表无效。

措施

更正创建索引命令中的高速缓存表列列表。确保指定的表中存在这些列。如果对列应用了函数，那么验证是否正确地使用了函数。

CTE0206: ATTRIBUTE 表达式无效。

说明

属性表达式中的列列表无效。

措施

更正创建索引命令中的属性列列表。确保指定的表中存在这些列。如果对列应用了函数，那么验证是否正确地使用了函数。

CTE0207: 未对视图“%1”.“%2”的索引指定 KEY COLUMNS FOR INDEX ON VIEW。

说明

如果创建了视图的索引，那么必须指定 KEY COLUMNS FOR INDEX ON VIEW(SQL-columname-list) 子句。列名列表指定“唯一地”标识视图中的某行的各列。

措施

将 KEY COLUMNS FOR INDEX ON VIEW(SQL-columname-list) 子句包括在创建索引命令中。

CTE0208: INITIAL SEARCH RESULT ORDER 列无效。

说明

INITIAL SEARCH RESULT ORDER(SQL-order-by list) 表达式中的列列表无效。

措施

更正创建索引命令中的排序列列表。检查语法是否正确以及指定的表中是否存在这些列。如果对列应用了函数，那么验证是否正确地使用了函数。

CTE0209: 属性列“%2”的类型“%1”不受支持，类型 DOUBLE 是必需的。

说明

对于属性列，唯一受支持的数据类型是 DOUBLE。

措施

确保具有要建立索引的文本列的表的属性列类型为 DOUBLE。可在属性列表达式中使用强制类型转换运算符。有关可强制类型转换为双精度的数据类型，请参阅 SQL Reference。

CTE0210: 索引配置参数“%2”的值“%1”无效。有效值为“%3”。

说明

配置参数的指定值不正确。有关参数的有效值，请参阅命令语法。

措施

更正创建索引命令中的索引配置参数值。

CTE0211: “%1”是无效的索引配置参数。

说明

索引配置选项未知。

措施

检查创建索引命令语法。有效的索引配置选项包括 TreatNumbersAsWords 和 IndexStopWords。下列各项必须用逗号隔开: index configuration(treatnumberaswords 1, indexstopwords 1)。

CTE0212: 未能保存内部索引配置文件“%1”。

说明

未能保存索引的内部配置文件。

措施

确保实例所有者对应保存文件的目录具有写许可权。如果同名文件已存在，那么确保它对于实例所有者是可写的。

CTE0213: 未能装入内部索引配置文件模板“%1”。

说明

未能读取内部索引配置文件模板。

措施

确保文件存在于正确的位置，并且是可读的。

CTE0214: 对索引配置文件设置新条目“[%1],%2=%3”时发生内部错误。

说明

编写索引的内部配置文件时发生内部错误。

措施

如果文件存在，那么检查对于实例所有者它是否为可读/写的。检查文件所在的设备上是否有足够的空间。

CTE0215: 不支持对别名“%1”.“%2”创建索引。使用基本表“%3”.“%4”来代替。

说明

不能对别名创建索引。

措施

输入带有基本表的创建索引命令。

CTE0217: 调度服务已经处于活动状态。

说明

服务已经是活动的，不需要启动。

措施

不需要任何操作。

CTE0218: 函数“%1”失败，错误代码为“%2”。

说明

Windows 函数失败，出现指定错误代码，不允许进一步处理。

措施

使用指定的 Windows 系统错误代码来获取详细的错误信息。

CTE0219: 未能打开服务“%1”。错误代码为“%2”。

说明

在 Windows 系统上找不到指定的服务。

措施

检查在 Windows 系统上是否安装了指定服务。使用指定的 Windows 系统错误代码来获取详细的错误信息。

CTE0220: 找不到 DB2 实例概要文件路径。

说明

用来获取 DB2 实例概要文件路径的内部 DB2 函数失败。

措施

创建 DB2 实例而不指定实例概要文件路径信息并重试命令。

CTE0221: 未正确指定 UpdateFrequency“%1”。

说明

更新频率语句的语法不正确。

措施

按照语法规范更正更新频率语句。

CTE0222: 调度服务输入文件“%1”已毁坏。

说明

包含索引更新信息的调度程序文件已毁坏。

措施

使用系统编辑器并尝试更正问题。可能已截断条目，或者已删除结束行字符。如果这样做不能复原文件内容，那么尝试以下操作:

- 调用命令 `db2text stop` 来停止调度程序。
- 删除调度程序服务文件。
- 调用 `db2text start` 命令来启动调度程序。
- 使用 `db2text alter index ...` 命令来重新创建所有相关索引的更新频率条目。

CTE0223: 未能关闭文件“%1”。

说明

不能关闭指定的文件。

措施

验证是否正确地指定了文件。

CTE0224 : 未能将文件“%1”复制至“%2”。

说明

不能将第一个文件复制至第二个文件。

措施

验证是否正确地指定了文件。检查第二个文件是否已存在并且是只读的。还应检查系统上是否有足够的可用空间。

CTE0225: 未能除去文件“%1”。

说明

不能从系统除去指定的文件。

措施

验证是否正确地指定了文件并检查文件访问权。

CTE0227: 对文件“%1”的写操作失败。

说明

指定的文件不是可写的。

措施

验证是否正确地指定了文件并检查文件文件访问权。还应检查系统上是否有足够的可用空间。

CTE0228: 用户在操作系统级别没有足够的访问权。

说明

命令需要操作系统级别的管理员权限。

措施

确保您具有操作系统管理员权限。检查您是否是管理员组的成员。

CTE0231: 未在表空间“%2”.“%3”所在的节点组“%4”中定义“%1”。

说明

管理表的表空间需要分布在不同的节点之间，分布方式与包含要建立索引的文本列的表的分布方式完全相同。为了强制执行此操作，会检查指定的表空间是不是在同一节点组中定义的。

措施

指定在包含要建立索引的文本列的表所在的节点组中定义的表空间。

CTE0232: 所指定或缺省表空间“%1”不是单节点表空间。这对于视图的索引或在指定 CACHE TABLE 选项时是必需的。

说明

对于单个节点上的表，只有视图的索引或在启用了 CACHE TABLE 选项的情况下受支持。

措施

如果缺省表空间导致此错误，那么将表放置在单节点表空间中。或者，如果指定了多节点表空间，那么指定另一单节点表空间。

CTE0233: 正在运行的管理命令存在冲突。请稍后重试此命令。

说明

另一管理命令仍在运行或异常终止而没有释放命令锁定。

措施

使用 CONTROL LIST 检查哪些锁定仍是活动的。如果存在活动锁定而没有任何命令正在运行，那么使用 CONTROL CLEAR 命令手动清除锁定。注意其他用户可能在运行持有锁定的管理命令。

CTE0234: 正在对文本索引运行的管理命令存在冲突。请稍后重试此命令或指定 DISABLE DATABASE 命令的 FORCE 选项。

说明

另一管理命令仍在运行或异常终止而没有释放命令锁定。

措施

使用 CONTROL LIST 检查哪些锁定仍是活动的。如果存在活动锁定而没有任何命令正在运行，那么使用 CONTROL CLEAR 命令手动清除锁定。注意其他用户可能在运行持有锁定的管理命令。对于 DISABLE DATABASE 命令，可以指定将停止对该数据库的所有其他命令的 FORCE 选项。

CTE0235: 找不到 Net Search Extender 的任何有效许可证。

说明

找不到 Net Search Extender 的任何有效许可证。

措施

检查是否使用 `db2lic` 正确地安装了许可证。确保在产品安装之后更新了现有实例。

CTE0236: 对于 MPP 实例，只有 Node0 是受支持的。

说明

如果具有要建立索引的文本列的表位于 Node0 上，那么只能对 MPP 实例创建文本索引。

措施

检查定义表的表空间的节点组。

CTE0237: 内部错误：日志表“%1”.“%2”包含无效操作“%3”。

说明

日志表记录对包含已索引文本列的表执行的操作。此表包含并非由 Net Search Extender 写下的条目，它可能已被毁坏。

措施

检查日志表，并删除毁坏的条目。

CTE0238: 内部错误：表“%1”.“%2”的列“%3”中包含不正确的语法表达式。

说明

表达式列表中的指定文本列存在错误。

措施

检查定界符“Begin”和“End”对。

CTE0239: 内部错误：索引属性“%1”的总长度超出最大长度“%2”。

说明

超出索引属性的最大大小（1016 字节）。属性包含实例、索引、工作目录以及其他信息。

措施

确保这些路径名不会过长。

CTE0240: 内部错误：设置环境变量“%1”失败。

说明

设置指定的环境变量失败。环境设置可能存在问题。

措施

检查特定于 OS 的准则。

CTE0242: 参数“%2”的值“%1”无效。

说明

搜索存储过程或表值函数 `DB2EXT.TEXTSEARCH` 是使用无效参数调用的。

措施

更正搜索存储过程或表值函数的参数值。有关有效的参数，请参阅文档。

CTE0243: 文本索引“%1”.“%2”尚未被激活。

说明

Net Search Extender 操作需要激活的高速缓存。高速缓存当前未激活。可能的原因包括:

- 在上一次 DB2TEXT START 命令之后从未激活高速缓存。
- 已使用 DB2TEXT DEACTIVATE CACHE 命令显式取消激活了高速缓存。

措施

对索引执行 DB2TEXT ACTIVATE CACHE 命令，并重新运行 Net Search Extender 操作。

CTE0244: 内部错误：对“%1”的调用返回 rc=″%2″, SQLCODE=″%3″。

说明

调用内部函数时发生内部处理错误。

措施

如果错误仍然存在，那么启动跟踪并检查 `db2diag.log`。报告错误。

CTE0245: 请求的高速缓存大小超出可用的高速缓存大小。将最大高速缓存大小增加至大于“%1”的值或降低 pctfree 值。

说明

装入所有数据所需的高速缓存大小超出索引的 MAXIMUM CACHE SIZE 值。在激活高速缓存（DB2TEXT ACTIVATE 命令）期间或在激活高速缓存时由索引更新操作检测到这一点。

措施

如果在 DB2EXT ACTIVATE 命令中报告了错误，那么使用 DB2EXT.MAXIMUM_CACHE_SIZE 函数重新计算最大高速缓存大小，并改变索引的 MAXIMUM CACHE SIZE 设置。最后降低 PCTFREE 值。如果增量更新期间超出最大文档数，那么使用 `db2 deactivate cache` 和 `db2text activate cache recreate` 命令重建高速缓存。

CTE0246: 文件“%1”为空。

说明

DB2TEXT CREATE INDEX 命令失败，原因是在命令中指定的文档模型文件是空的。

措施

在命令中指定有效的文档模型文件。

CTE0247: 未能创建 Net Search Extender 存储过程。

说明

DB2TEXT ENABLE DATABASE 命令未能创建内部存储过程 DB2EXT.CTESRVSP。

措施

有关详细信息，检查与 CREATE PROCEDURE 语句相关联的其他 DB2 错误消息。如果通过除去具有完全相同的名称的现有存储过程不能更正错误，那么启动跟踪并报告错误。

CTE0248: 生成的搜索字符串过长。请降低搜索查询的复杂性。

说明

Net Search Extender 查询太长或太复杂，基本搜索引擎无法对其进行处理。复杂性受同义字比较、FUZZY FORM OF 表达式和屏蔽字符的影响。

措施

降低查询的复杂性或长度。

CTE0249: 可执行程序“%1”异常终止。

说明

执行 Net Search Extender 命令时，调用了可执行文件“%1”，但它异常终止了。

措施

验证可执行文件是否尚未通过用户交互作用（例如，信号）显式终止。如果没有，那么启动跟踪，重新运行命令，并报告错误。

CTE0250 ：列类型变换函数“%2”.“%3”的返回类型“%1”不受支持。

说明

在 DB2TEXT CREATE INDEX 命令中，指定了返回不受支持的数据类型的列类型变换。受支持的数据类型包括：CHARACTER、VARCHAR、LONG VARCHAR（不推荐）、CLOB、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC（不推荐）、DBCLOB、BLOB 和 XML。

措施

选择不同的列类型变换函数。

CTE0251: 内部错误：列类型“%1”不受支持。

说明

使用的列类型不在受支持类型的列表中。

措施

检查创建索引以获取“键和建立索引”的有效列的列表。对命令作适当更改并重试。如果仍然存在错误，那么启动跟踪并检查 db2diag.log。向 IBM 服务中心报告错误。

CTE0252: 缺少参数“%1”。

说明

内部错误 － 执行 Net Search Extender 命令时，在缺少参数“%1”的情况下调用管理可执行程序。

措施

尝试更改 Net Search Extender 参数命令来避免发生问题。如果错误仍然存在，那么打开跟踪功能，并向 IBM 服务中心报告错误。

CTE0253: 找不到日志视图中列示的文档。

说明

日志视图中列示的文本文档的内容已更改，未能访问这些内容。

措施

检查文档是否存在以及要包括在索引中的文本文档的读取/访问许可权。

CTE0254: 索引“%1”的高速缓存已激活。

说明

已使用 ACTIVATE CACHE 命令激活索引。

措施

检查指定的索引名和正在使用的数据库。

CTE0255 ：缺少高速缓存结果列表达式的列名。将“ AS <cache column name>”添加至表达式。

说明

必须命名高速缓存结果列表达式。例如: “C1+C2 AS myresult”。

措施

将“AS <cache column name>”添加至表达式。

CTE0256: 为建立索引选择数据所需的查询失败。请降低属性、高速缓存表或初始搜索结果顺序表达式的复杂性。

说明

Net Search Extender 从命令的表达式创建查询以便为建立索引而从数据库选择数据。查询失败，原因是它太复杂。

措施

降低属性、高速缓存表或初始搜索结果顺序表达式的复杂性。

CTE0257: 创建共享内存时出错。

说明

由于先前的错误或许可权问题，未能创建共享内存资源。

措施

检查 `db2diag.log` 以获取更多信息，或者清除共享资源。另请参阅错误 CTE0183。

CTE0258: 共享内存版本错误。

说明

未能访问共享内存资源，原因是它已毁坏或存在版本冲突。

措施

有关进一步信息，检查 `db2diag.log`。禁用并重新启用数据库，然后重试。

CTE0259: 不能在全局共享内存中插入条目。条目已存在。

说明

由于先前的错误，要在全局共享内存中插入的条目已存在。

措施

有关进一步信息，检查 `db2diag.log`。使用 `db2text stop` 和 `db2text start` 命令重新启动更新和锁定服务。

CTE0260: 不能访问全局共享内存中的条目。找不到条目。

说明

由于先前的错误，要从全局共享内存中除去的条目不存在。

措施

有关进一步信息，检查 `db2diag.log`。尝试使用 `db2text stop` 和 `db2text start` 命令重新启动更新和锁定服务。

CTE0261: 对于此实例中的文本索引，至少存在一个激活的高速缓存。使用 DEACTIVATE CACHE 命令取消激活任何已激活的索引的高速缓存，或者使用 FORCE 选项来停止。

说明

仅当对使用 ACTIVATE CACHE 命令激活的所有文本索引运行 DEACTIVATE CACHE 命令时，才能使用 db2text stop 命令。

措施

使用 DEACTIVATE CACHE 命令取消激活任何已激活的索引的高速缓存或使用 FORCE 选项来停止。

CTE0262: 参数“%1”的值过长。

说明

值超过最大允许大小。

措施

检查最大大小。

CTE0263: 使用 RECREATE INDEX ON UPDATE 选项创建了文本索引“%1”.“%2”。在此上下文中，可以不指定 UPDATE MINIMUM 或 COMMITCOUNT FOR UPDATE。

说明

仅当以增量方式更新索引时，update minimum 和 commitcount for update 才有效。

措施

如果想要每次执行更新时都重新创建索引，那么除去 UPDATE MINIMUM 和 COMMITCOUNT FOR UPDATE 设置。如果想要使用 UPDATE MINIMUM 和 COMMITCOUNT FOR UPDATE，那么不要指定 RECREATE INDEX ON UPDATE。

CTE0264: 激活索引操作中发生了错误。请检查事件视图“%1”.“%2”和 db2diag.log 以了解详细信息。

说明

在索引激活过程期间，错误会写至事件表和 `db2diag.log` 文件。

措施

有关文档错误的更多信息，检查事件表。在修正问题之后清除事件日志。

CTE0265: 用户表或管理表空间（“%1”）的表空间不只是对节点 0 定义的。

说明

如果文本索引是对 MPP 实例创建的，那么用户表的表空间一定只位于 Node0 上。

措施

使用表空间位于 Node0 上的表。

CTE0266: ValueFrom“%1”必须小于 ValueTo“%2”。

说明

属性搜索中指定的值无效。如果搜索语法为“BETWEEN ValueFrom AND ValueTo”，那么下边界（ValueFrom）必须小于上边界（ValueTo）。

措施

更改“BETWEEN ValueFrom AND ValueTo”子句中的边界。

CTE0267: 数据库“%1”中的 Net Search Extender 数据库对象处于不一致状态。

说明

至少缺少或毁坏了一个 Net Search Extender 对象。在安装新的 Net Search Extender 产品版本之后尚未迁移数据库，或者数据库用户已更改或删除 Net Search Extender 内部对象。在这种情况下，所有文本索引都会丢失，数据库就不能再使用文本。

措施

要将数据库迁移至当前版本，请遵循 Net Search Extender 文档中的迁移描述。或者，使用 FORCE 选项发出 DB2TEXT DISABLE DATABASE 命令。然后，您可以通过使用 DB2TEXT ENABLE DATABASE 命令再次启用数据库的文本搜索。

CTE0270: 无法在增量更新后修改日志表“%1”.“%2”。下一次更新（UPDATE）期间要处理条目。

说明

启动增量索引更新时，会创建时间戳记。它充当要处理的更改记录的阈值。于是与增量更新同时发生的更改将在以后进行处理（在下一次更新期间）。在某些情况下，更新启动时未落实但在执行索引更新时落实的事务中可能有一些更改。这有可能导致不一致。

为了避免出现这种不一致的情况，不要从日志表中删除阈值时间戳记之前的更改记录，即使它们只作了部分处理。在下一次增量更新时，将对索引重新应用这些更改。

措施

在下次索引更新时，会对索引重新应用这些更改。对于删除操作，这可能导致以下错误：CTE0101: ItlEnReasonCode_Docmap_docid_not_found。

注意，可忽略此错误，原因是已删除文档。如果 CTE0270 错误经常发生，那么考虑删除索引并使用增量索引更新的已修改时间戳记阈值重新创建索引。例如：db2text ″CREATE INDEX ... INDEX CONFIGURATION(UPDATEDELAY 30)″

这意味着增量更新期间处理只运行早于 30 秒的更改记录，而避免干预少于 30 秒的并发更改事务。

CTE0271: 不可使用高速缓存，需要执行 DEACTIVATE 和 ACTIVATE RECREATE。

说明

因为已经达到最大高速缓存大小，所以高速缓存处于不一致状态。

措施

检查最大高速缓存大小是否仍然够用。然后调用下列 db2text 命令：DEACTIVATE CACHE 和 ACTIVATE CACHE RECREATE。

CTE0272: 高速缓存大小不是足够大。增大 PCTFREE 值，或者使用 DEACTIVATE 和 ACTIVATE [RECREATE] 来重新创建高速缓存。

说明

已经使用了为高速缓存保留的所有内存。

措施

通过使用以下顺序的 db2text 命令来重建高速缓存：DEACTIVATE CACHE、ALTER INDEX MAXIMUM CACHE SIZE 和 ACTIVATE CACHE RECREATE。

CTE0273: 索引“%1”，“%2”的高速缓存已激活。

说明

已使用 ACTIVATE CACHE 命令激活索引。

措施

检查指定的索引名和正在使用的数据库。

CTE0274: 连接的目标数据库系统“%1”不受支持。

说明

已尝试在与 Net Search Extender 不支持的数据库系统建立了连接的情况下执行 DB2TEXT 命令。

CTE0275: 找不到服务器“%1”的类型和版本信息。

说明

在 DB2 目录视图“SERVERS”中找不到服务器的类型和版本信息。

措施

确保正确设置了 DB2 联合环境。

CTE0277: 未能连接高速缓存存储段。

说明

系统不能分配足够的内存来装入大型高速缓存段或打不开高速缓存段（原因是先前已删除它）。

措施

检查系统设置并增加调页空间和可用内存量。对于大型高速缓存大小，您可能需要准备系统。请参阅 Net Search Extender 文档。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。如果问题仍然存在，那么检查 db2diag.log 以获取更多信息。

CTE0278: 在 AIX 32 位系统上，在激活大型高速缓存之前更改 MAXDATA 设置。

说明

当在 AIX 32 位系统上使用搜索存储过程时，可能需要更改 db2fmp 可执行文件的 MAXDATA 设置。

措施

有关更改 MAXDATA 设置的详细信息，请参阅 Net Search Extender 文档。

CTE0279: 高速缓存数据的大小已达到系统限制。

说明

通过降低 PCTFREE 值，可以增加高速缓存激活期间的最大数据大小。这使系统能够在高速缓存中保留较少的可用空间。

措施

使用较低的 PCTFREE 值或减少要高速缓存的文本数据量。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。

CTE0280: 没有足够的磁盘空间来写入持久高速缓存文件。

说明

系统不能在高速缓存目录中为持久高速缓存写入足够大的文件。

措施

使用 ALTER INDEX 命令，将持久高速缓存目录更改为空的文件系统。或者通过降低 PCTFREE 或 MAXIMUM CACHE SIZE 值或通过使用临时高速缓存来减少高速缓存大小。

CTE0281: 删除持久高速缓存文件“%1”失败。

说明

文件不存在或不能访问。

措施

检查此文件是否仍然存在并手动删除。

CTE0282: 高速缓存中的文档数已达到系统限制。

说明

通过降低 PCTFREE 值，可以增加高速缓存激活期间要高速缓存的最大文档条目数。这使系统能够在高速缓存中保留较少的可用空间。

措施

使用较低的 PCTFREE 值或减少高速缓存中的文档条目的量。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。

CTE0283: 未能创建高速缓存存储段。

说明

系统不能为将大型高速缓存段装入到内存中分配足够内存。通过降低 PCTFREE 值，可达到较小的高速缓存段大小。

措施

检查系统设置并增加调页空间和可用内存量。可以通过使用较低的 PCTREE 值来降低高速缓存大小。对于大型高速缓存大小，可能需要准备系统。请参阅 Net Search Extender 文档。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。如果问题仍然存在，那么检查 db2diag.log 以获取更多信息。

CTE0284: 文本索引位于节点“%1”上，但是对节点“%2”调用了搜索函数。

说明

未对索引所在的节点调用搜索存储过程或表值函数 DB2EXT.TEXTSEARCH。搜索函数将不会自动分布至正确节点。

措施

将 DB2NODE 环境变量设置为在连接数据库之前连接索引的节点。

CTE0285: 对于分布至多个节点的文本索引，不允许搜索函数。

说明

一定不要使用分布至多个节点的索引调用表值函数 DB2EXT.TEXTSEARCH，原因是它将不会自动分布至正确节点，而是在协调程序节点上执行。

措施

在多节点环境中使用 CONTAINS、SCORE 或 NUMBEROFMATCHES 函数。

CTE0286: 未在源表"%2"."%3"和捕获更改表"%4"."%5"的"%1"."IBMSNAP_REGISTER"找到行。

说明

对于在 DB2TEXT CREATE INDEX 命令中指定的复制捕获表特征，在 IBMSNAP_REGISTER 表中找不到任何有效条目。对于索引列 SOURCE_OWNER 和 SOURCE_NAME（SOURCE_VIEW_QUAL=0）以及列 PHYS_CHANGE_OWNER 和 PHYS_CHANGE_TABLE 中的指定复制捕获表，有效条目必须包含指定源表。

可能原因：指定源表未注册为复制捕获表的复制源。

措施

为 DB2 复制正确注册源表，或为源表指定正确的复制捕获表。

CTE0287: 源表"%4"."%5"和捕获更改表"%6"."%7"的"%3"."IBMSNAP_REGISTER"中"%2"的值"%1"无效。

说明

不允许 IBMSNAP_REGISTER 表中的复制设置。可能原因：1. 列 CHG_UPD_TO_DEL_INS 未包含值"Y"。2. 列 CCD_CONDENSED 包含值"Y"。

措施

当为 DB2 复制注册源表时，确保将更新操作变换为删除和插入操作对。此外，还应确保没有使用任何压缩复制捕获表。

CTE0288: 源表"%1"."%2"和捕获更改表"%3"."%4"位于不同的服务器（"%5"和"%6"）上。

说明

指定的源表和复制捕获表必须位于同一服务器上。

CTE0289: 包装器"%1"不受支持。

说明

包装器不受支持。有关受支持包装器的列表，请参阅 Net Search Extender 文档。

CTE0290: 别名“%1”.“%2”在复制子句中未被允许。

说明

不允许在复制子句中对昵称指定别名。

措施

指定昵称而不是别名或为远程表创建新昵称。

CTE0291: 不允许对类型为 XML 的列使用指定的格式。

说明

对于 XML 列，只允许使用“XML”格式。

措施

指定“XML”格式或者不指定任何格式。

CTE0292: 捕获到 Windows 异常“%1”，地址=“%2”，标志=“%3”。

说明

发生了 Windows 异常。提供了异常名称、地址和标志。

CTE0293: 捕获到 Windows 异常“%1”。

说明

发生 Windows 异常。

CTE0294: 发生了搜索自变量处理问题。

说明

由于环境设置不正确，因此发生了搜索自变量处理错误。

措施

检查语言环境 charmap 值是否与 DB2 代码页相匹配以及在系统上是否可用。

CTE0295: 为非二进制文本列指定了无效 CCSID“%1”。

说明

对于非二进制文本列数据类型，DB2 始终会将数据存储在数据库 CCSID 中。仅数据库 CCSID 对于非二进制文本列有效。

措施

省略 CCSID 子句或者指定有效的 CCSID。

CTE0296: 在“%2”上找不到库“%1”。请检查 Net Search Extender 安装。

说明

您试图使用 DB2 控制中心来执行 Net Search Extender 命令。而在目标系统上未正确安装 Net Search Extender。

措施

检查在目标系统上是否正确安装了 Net Search Extender。

CTE0297: 数据库“%1”与 DB2 文本搜索相关联

说明

不允许将数据库与多于一个文本搜索组件相关联。

措施

如果要继续将 DB2 文本搜索与数据库使用配合使用，那么无需任何操作。如果选择使用 DB2 Net Search Extender，那么通过 DB2 文本搜索禁用数据库，然后重试此命令。

CTE0298: 数据库“%1”已与 DB2 Text Search 相关联。无法执行此命令。

说明

不允许将数据库与多于一个文本搜索组件相关联。

措施

如果不想使用 DB2 Net Search Extender，那么使用 DB2TEXT DISABLE DATABASE 命令禁用数据库。如果选择使用 DB2 Net Search Extender，那么通过 DB2 Text Search 禁用数据库，然后重试此命令。

CTE0360: 显示了特定错误消息

说明

此消息号是为特定 Net Search Extender 错误保留的。

措施

对特定错误消息作出响应。

错误消息 CTE0451 - CTE0866

本节包含 Net Search Extender 生成的错误消息

CTE0451: 突出显示 UDF 不支持指定的文档格式“%1”。

说明

文档格式“%1”不支持突出显示。

措施

使用突出显示 UDF 支持的文档格式。

CTE0452: 突出显示 UDF 中的选项“%1”附近存在语法错误。

说明

在指定选项附近指定了不正确的语法。

措施

检查选项 %1 附近的语法。更正错误并重试。

CTE0453: 突出显示 UDF 的返回大小过小。

说明

突出显示文档的请求部分无法装入到突出显示 UDF 的返回参数中。

措施

降低窗口编号、窗口大小和/或应该显示点击的节数。这将减少返回给用户的文档部分。

CTE0454: 将突出显示 UDF 的参数从代码页“%1”转换为代码页 UTF8 时发生了错误。

说明

指定 CCSID（它可能是缺省数据库 CCSID）中的突出显示 UDF 的参数不能转换为 UTF8。

措施

确保正确地指定 CCSID。

CTE0455: 数据库代码页“%1”在突出显示 UDF 中不受支持。

说明

突出显示 UDF 不支持数据库的代码页。

CTE0456: 突出显示 UDF 只支持代码页为 UTF8 的文档。

说明

只有代码页为 UTF8 的文档才支持突出显示 UDF。

CTE0457: 参数“%2”的值“%1”在突出显示 UDF 中无效。

说明

突出显示参数的值无效。

措施

检查参数值，并确保在数据范围内允许该值。

CTE0458: 用法: db2exthl <new size in kilo bytes>。

说明

db2exthl 实用程序的参数不正确。

措施

请提供一个 1 到 1048576 之间的值。

CTE0841: 缺少命令选项“%1”。

说明

未指定必需的命令选项。

措施

请检查指定的参数并添加缺少的参数。

CTE0842: 没有为命令选项“%1”指定任何值。

说明

未指定命令选项的必需值。

措施

请检查指定的参数并添加缺少的选项。

CTE0843: 没有为命令选项“%1”指定任何数字值。

说明

指定了字符串而不是数字。

措施

请检查指定的参数并将字符串更改为正确的数字。

CTE0844: 定义文件路径“%1”过长。

说明

指定的路径过长因而未能处理。

措施

请使用较短的路径然后重试。

CTE0845: 没有指定任何定义文件。

说明

需要指定定义文件。

措施

添加有效定义文件并再次尝试调用。

CTE0846: 定义文件名“%1”过长。

说明

指定的定义文件名过长。

措施

将定义文件名的长度减少为允许的大小。

CTE0847: 定义文件“%1”不存在。

说明

找不到指定的定义文件。

措施

检查定义文件是否在正确的路径中以及是否可供当前用户访问。

CTE0849: 未能锁定字典文件“%1”。

说明

进程无法锁定字典文件。您没有写访问权或另一进程已打开文件以进行写操作。

措施

检查运行的进程以确保没有任何进程正锁定字典文件，并检查您的访问权。

CTE0850: 输出文件“%1”已存在。

说明

未能覆盖指定的输出文件。

措施

检查您能否在指定目录中创建同义字。

CTE0851: 字典文件“%1”不完整。

说明

同义字典文件已毁坏。

措施

清除目录并再次编译定义文件。

CTE0852: 字典文件“%1”版本错误。

说明

字典文件是使用同义字编译器的较早版本生成的。

措施

使用同义字编译器的最新版本再次编译定义文件。

CTE0853: 不能覆盖现有字典“%1”。

说明

不能覆盖现有字典。

措施

检查对字典文件的写访问权、它的目录位置和子目录位置。

CTE0855: 未正确指定同义字词条。

说明

定义文件中存在语法错误。

措施

有关创建同义字定义文件和同义字支持的信息，请检查 Net Search Extender 文档。

CTE0856: 定义文件“%1”为空。

说明

不允许空的定义文件。

措施

有关创建同义字定义文件和同义字支持的信息，请检查 Net Search Extender 文档。

CTE0857: 在文件“%1”的第“%2”行中找不到任何块起始行。

说明

定义文件中存在语法错误。

措施

块必须以“:WORDS”开始。有关同义字概念的信息，请检查 Net Search Extender 文档。

CTE0858: 在文件“%1”的第“%2”行中指定了无效关系。

说明

定义文件中存在语法错误。

措施

必须检查“相关联的词条定义”。有关创建同义字定义文件的信息，请检查 Net Search Extender 文档。

CTE0859: 关系编号不在文件“%1”的第“%2”行的范围内。

说明

用户定义的关系都是以相关联的类型为基础的。它们由 1 与 128 之间的唯一数字标识。

措施

验证关系编号。

CTE0860: 没有在文件“%1”的第“%2”行中定义任何词条。

说明

未指定必需词条。

措施

有关创建同义字定义文件的信息，请检查 Net Search Extender 文档。

CTE0861: 文件“%1”的第“%2”行中的同义字词条过长。

说明

同义字词条的长度限制为 64 个字节。

措施

改变同义字词条的大小并重试。

CTE0862: 在文件“%1”的第“%2”行中错误指定了强度。

说明

定义文件中存在语法错误。

措施

有关创建同义字定义文件和同义字支持的信息，请检查 Net Search Extender 文档。

CTE0863: 文件“%1”的第“%2”行中的强度超出范围。

说明

强度值应指定为 1 与 100 之间。

措施

更改强度值以使它为从 1 至 100 的数字值。

CTE0864: 内部错误: 同义字编译器失败，原因码为“%1”。

说明

发生了内部处理错误，不允许进一步处理。尝试启动并停止更新和锁定服务以及 DB2。

措施

如果错误仍然存在，那么启动跟踪并检查 db2diag.log。

CTE0865: 未能创建字典“%1”。

说明

未能创建指定的目录。

措施

检查目录是否已经存在以及针对该目录的许可权。

CTE0866: 未能除去字典“%1”。

说明

未能除去目录。

措施

检查您是否对指定的目录具有写许可权。

Windows 系统错误

以下是 Windows 系统错误的列表:

系统错误

1 函数不正确。

2 系统找不到指定的文件。

3 系统找不到指定的路径。

4 系统打不开文件。

5 拒绝访问。

6 句柄无效。

8 没有足够的存储空间可用来处理此命令。

14 没有足够的存储空间可用来完成此操作。

15 系统找不到指定的驱动器。

29 系统不能写入指定的设备。

30 系统不能从指定的设备读取。

32 进程不能访问该文件，原因是它正由另一个进程使用。

36 打开太多的文件进行共享。

38 已到达文件的末尾。

39 磁盘已满。

80 文件已存在。

82 不能创建目录或文件。

100 不能创建另一个系统信号量。

101 专用信号由另一个进程所拥有。

102 设置了信号并且不能将它关闭。

103 不能再次设置该信号。

104 不能在中断时请求专用信号。

105 此信号的前一个所有权已结束。

110 系统打不开指定的设备或文件。

111 文件名太长。

112 磁盘上没有足够空间。

121 信号超时这一时间段已到期。

126 找不到指定的模块。

127 找不到指定的过程。

147 没有足够的资源可用来处理此命令。

155 不能创建另一线程。

161 指定的路径无效。

164 不能在系统中再创建其他线程。

170 请求的资源在使用中。

183 当文件已存在时不能创建该文件。

187 找不到指定的系统信号量名称。

206 文件名或扩展名太长。

267 目录名无效。

288 尝试释放并非由调用者所拥有的互斥。

298 对信号提出了太多公布要求。

998 对内存位置的访问无效。

1051 停止控制权已发送至其他运行的服务所依赖的服务。

1052 请求的控制权对此服务无效。

1053 服务未及时响应启动或控制请求。

1054 未能为服务创建线程。

1055 服务数据库已锁定。

1056 服务的实例已在运行。

1057 帐户名无效或不存在。

1058 不能启动服务，原因是已禁用该服务或它没有与服务相关联的已启用设备。

1059 指定了循环服务依赖关系。

1060 指定的服务不是作为已安装的服务存在。

1061 此时服务不能接受控制消息。

1062 服务尚未启动。

1063 服务进程未能连接至服务控制器。

1064 处理控制请求时服务中发生了异常。

1066 服务已返回特定于服务的错误代码。

1067 进程意外终止。

1068 依赖关系服务或组未能启动。

1069 因为登录失败，所以服务未启动。

1070 启动之后，服务处于启动暂挂状态。

1071 指定的服务数据库锁定无效。

1072 指定的服务已标记为删除。

1073 指定的服务已存在。

1078 该名称已用作服务名称或服务显示名。

1079 为此服务指定的帐户不同于为在同一进程中运行的其他服务指定的帐户。

1082 未对此服务配置任何恢复程序。

1154 运行此应用程序所需的一个库文件已破坏。

1219 提供的凭证与现有的一组凭证冲突。

1242 已注册该服务。

1243 指定的服务不存在。

1244 未执行正在请求的操作，原因是尚未认证用户。

1245 未执行正在请求的操作，原因是用户尚未登录网络。指定的服务不存在。

1392 文件或目录已毁坏且不是可读的。

1455 页面调度文件太小，此操作无法完成。

1793 用户帐户已到期。

Net Search Extender 信息目录

Net Search Extender 将有关缺省值、配置、文本索引和格式的重要信息存储在目录表中。要查看此信息，可对表查询一些视图。

下列视图和表反映系统的当前配置:

- 数据库级别信息视图:
  - db2ext.dbdefaults
- 索引级别信息视图:
  - db2ext.textindexes
  - db2ext.textindexformats
  - db2ext.indexconfiguration
- 文本索引的表视图:
  - 事件视图
  - 日志表视图

数据库级别信息的视图

db2ext.dbdefaults 视图显示数据库的所有缺省值。

数据库级别上的缺省值不能更改，并且在此视图中以“属性/值”对的形式提供:

```
db2ext.dbdefaults

db2 select DEFAULTNAME, DEFAULTVALUE from DB2EXT.DBDEFAULTS
```

表 8. db2ext.dbdefaults 视图

属性	缺省值	注释
CCSID	数据库的 CCSID	文档的缺省 CCSID。如果在 CREATE INDEX 命令中未指定任何 CCSID，那么将它应用。
FORMAT	TEXT	文档缺省格式。如果在 CREATE INDEX 命令中未指定任何格式，那么将它应用。

表 8. db2ext.dbdefaults 视图（续）

属性	缺省值	注释
INDEXDIRECTORY	请参阅注释下面的路径名	全文本索引文件的目录。如果在 CREATE INDEX 命令中未指定任何索引目录，那么将它应用。路径名为: $DB2EXT_INSTOWNERHOMEDIR/sqllib/db2ext/indexes
LANGUAGE	EN_US	文档语言。
MODELCCSID	数据库的 CCSID	文档模型文件的 CCSID。
UPDATECOMMITCOUNT	0	更新期间在一个事务中处理的更改数。
CLEARCOMMITCOUNT	0	CLEAR INDEX 命令运行期间在一个事务中处理的更改数。
UPDATEFREQUENCY	NONE	何时检查新索引中的更新。
UPDATEMINIMUM	1	执行更新之前的最小更改数。
WORKDIRECTORY	请参阅注释下面的路径名	索引临时文件的目录。路径名为: <os_dependent>/sqllib/db2ext/indexes
CACHEDIRECTORY	请参阅注释下面的路径名	CREATE INDEX 命令的 PERSISTENT CACHE 选项的缺省目录。路径名为: <os_dependent>/sqllib/ db2ext/memory
PCTFREE	50	保留以供将来插入操作使用的高速缓存的百分比。
USERPERSISTENTCACHE	1	使用持久高速缓存。
AUTOMATICREORG	1	CREATE INDEX 命令中的 REORGANIZE 选项。这意味着自动重组。
TREATNUMBERSASWORDS	0	即使一系列字符和数字相邻，也不要将它们解释为独立的单词。例如，缺省值 0 意味着 tea42at5 被视作一个单词。
INDEXSTOPWORDS	1	为所有文本（包含停用词）创建索引。
VERSION		NSE V9.1 Net Search Extender 的当前版本号。
UPDATEDELAY	0	指定不使用捕获表的增量更新的持续时间（以秒计）。将仅从日志表中抽取早于此持续时间的条目。这是为了避免丢失更新。例如，在用户事务对更新命令有影响的事务情况中，未反映在索引中的文档更改。因此，UpdateDelay 参数应设置为对其创建索引的表的用户写事务的最长持续时间。

索引级别信息的视图

可以使用下列 Net Search Extender 视图查询索引级别的信息:

- db2ext.textindexes
- db2ext.textindexformats
- db2ext.indexconfiguration

- <index eventview name schema>.<index eventview name>

为了保持向后兼容性，DB2 Text Information Extender 视图 db2ext.textcolumns、db2ext.formats 和 db2ext.models 仍然受支持，但是建议不要使用。

注意：在 db2ext.textcolumns 视图中，OPERATION、OPERATIONBEGIN 和 OPERATIONEND 列不再受支持。

db2ext.textindexes 视图

对 Net Search Extender 启用的每个数据库包含 db2ext.textindexes 视图。这包含有关在此数据库中创建的文本索引的设置、统计信息和缺省值的信息。

当创建文本索引时，会在 db2ext.textindexes 中创建新的条目。当删除文本索引时，会删除这些条目。

可以查询视图以获取有关索引的信息。以下是使用索引模式的示例：

```
db2 "select COLNAME from DB2EXT.TEXTINDEXES where INDSCHEMA='myschema'
        and INDNAME='myindex'"
```

但要注意，不能使用常规 SQL 数据处理命令修改视图或显式创建或删除目录视图。视图的附加内容可在下表中找到。

还要注意，复制参数未包括在此视图中。

表 9. db2ext.textindexes 视图

属性	类型	注释
INDSCHEMA	VARCHAR(128)	文本索引的模式名。
INDNAME	VARCHAR(128)	文本索引的名称。
TABSCHEMA	VARCHAR(128)	基本表、昵称和视图的模式的表名。
TABNAME	VARCHAR(128)	对其创建索引的别名。
COLNAME	VARCHAR(128)	对其创建索引的列。
CCSID	INTEGER	此索引的文档 CCSID。
LANGUAGE	VARCHAR(5)	此索引的文档语言。
FUNCTIONSCHEMA	VARCHAR(128)	列映射函数的模式。
FUNCTIONNAME	VARCHAR(18)	列映射函数的名称。
INDEXDIRECTORY	VARCHAR(256)	全文本索引文件的目录。
WORKDIRECTORY	VARCHAR(256)	索引临时文件的目录。
CACHEDIRECTORY	VARCHAR(256)	如果 persistentcache=1，那么此属性表示持久高速缓存的目录。
UPDATEFREQUENCY	VARCHAR(300)	将自动更新应用于此索引的触发器条件。
UPDATEMINIMUM	INTEGER	在更新执行之前必须更改的最小文档数。
EVENTVIEWSCHEMA	VARCHAR(128)	为此索引创建的事件视图的模式。
EVENTVIEWNAME	VARCHAR(128)	为此索引创建的事件视图的名称。
LOGVIEWSCHEMA	VARCHAR(128)	为索引创建的日志视图的模式。
LOGVIEWNAME	VARCHAR(128)	为索引创建的日志视图的名称（对视图的增量更新很重要）。
COMMITCOUNT	INTEGER	落实计数更新的缺省值。

表 9. db2ext.textindexes 视图（续）

属性	类型	注释
NUMBER_DOCS	INTEGER	目前索引中的总文档数。注意，在索引更新期间，仅当设置 commitcount 时才更新此值。
REORG_SUGGESTED	INTEGER	指示通过运行 UPDATE INDEX REORGANIZE 是否能改进性能。仅当至少其中一个节点具有建议的索引重组时，此参数才为 true (1)。
REORGAUTOMATIC	INTEGER	如果在执行更新操作期间自动重组索引，那么此属性的值为 1。
RECREATEONUPDATE	INTEGER	如果在执行更新操作期间自动重组索引，那么此属性的值为 1。
CREATIONTIME	TIMESTAMP	索引创建的时间。
UPDATETIME	TIMESTAMP	上次更新的时间。如果 UPDATE TIME 等于 CREATION TIME，那么尚未处理任何更新。
PERSISTENTCACHE	INTEGER	如果使用持久高速缓存，那么此属性的值为 1。
MAXIMUMCACHESIZE	INTEGER	高速缓存的最大大小。
PCTFREE	INTEGER	保留以供将来插入操作使用的高速缓存的百分比。
CACHETABLE	VARCHAR(32000)	CACHE TABLE 的列表达式列表。
RESULTORDER	VARCHAR(32000)	INITIAL RESULT ORDER 的 SQL 排序依据。
ATTRIBUTES	VARCHAR(32000)	ATTRIBUTES 的列表达式列表。
VIEWKEYCOLUMNS	VARCHAR(32000)	视图的索引的键列。

db2ext.indexconfiguration 视图

索引配置参数可从 db2ext.indexconfiguration 视图中获取。可通过常规 SQL 查询工具获取该视图。以下是使用索引名的示例:

```
db2 "select VALUE from DB2EXT.INDEXCONFIGURATION where INDSCHEMA='myschema'
         and INDNAME='myindex' and PARAMETER ='INDEXSTOPWORDS'"
```

视图的附加内容可在下列表中找到。

表 10. db2ext.indexconfiguration 视图

属性	类型	注释
INDSCHEMA	VARCHAR(128)	索引的模式名。
INDNAME	VARCHAR(128)	索引的名称。
PARAMETER	VARCHAR(30)	参数的类型。
VALUE	VARCHAR(512	参数的值。

对于 PARAMETER 和 VALUE 属性，有几个值可用。

表 11. db2ext.indexconfiguration 视图

属性和值	属性和值
PARAMETER	VALUE
- TREATNUMBERASWORDS	- 0 或 1
- INDEXSTOPWORDS	- 0 或 1
- UPDATEDELAY	- 秒数大于等于 0

有关进一步信息，请参阅 CREATE INDEX 命令的 CONFIGURATION 选项。

db2ext.textindexformats 视图

可从 db2ext.textindexformats 视图中获取索引的格式和模型信息。以下是使用索引名的示例:

```
db2 "select FORMAT from DB2EXT.TEXTINDEXFORMATS where INDSCHEMA='myschema'
          and INDNAME='myindex'"
```

视图的附加内容可在下表中找到。

表 12. db2ext.textindexformats 视图

属性	类型	注释
INDSCHEMA	VARCHAR(128)	索引的模式名（用作日志表中的索引名和模式名的前缀）。
INDNAME	VARCHAR(128)	在 CREATE INDEX 命令中指定的索引名。
FORMAT	VARCHAR(30)	模型绑定至此格式。
MODELNAME	VARCHAR(30)	文档模型的名称。
MODELFILE	VARCHAR(256)	包含模型定义的文件。
MODELCCSID	INTEGER	MODELFILE 的 CCSID。
DEFAULT	INTEGER	目前为 1（原因是索引中的多个格式目前不受支持）。

文本索引的表视图

可以使用下列 DB2 Net Search Extender 视图来查询索引级别的信息:

- 事件视图
- 日志表视图

事件视图

此视图允许您获取有关建立索引状态、错误事件和建立索引期间何时发生问题（例如，找不到文档）的信息。然后，将这些索引更新事件写至索引的事件表。

此视图允许您获取有关建立索引状态、错误事件和建立索引期间何时发生问题（例如，找不到文档）的信息。然后，将这些索引更新事件写至索引的事件表。

模式和名称存储在 db2ext.textindexes 视图中。要获取事件视图的名称，使用以下示例:

```
db2 "select EVENTVIEWSCHEMA, EVENTVIEWNAME from DB2EXT.TEXTINDEXES
       where INDSCHEMA = 'myschema' and INDNAME = 'myindex'
```

索引的事件视图包含下列各列。

表 13. 事件视图

属性	类型	**Notes®**
OPERATION	INTEGER	对用户表的操作反映在全文本索引（insert = 0/ update = 1/ delete = 2）中。 当使用复制捕获表时，更新操作将分割为删除和插入操作。在此情况下，事件表中的插入操作可以来自于对其创建索引的源表的插入或更新操作。
TIME	TIMESTAMP	创建事件条目的时间戳记。
REASON	INTEGER	原因码。有关原因码的列表，请参阅第 207 页的『文本搜索引擎原因码』。
SEVERITY	INTEGER	表条目的严重性。例如，1 仅供参考，4 指示警告，而 8 意味着表条目错误。
MESSAGE	VARCHAR(1024)	附加文本信息。
KEY1, ... KEY14	依赖于用户表	用户表的第一个主键列至最后一个主键列（最大为 14）。
PARTITION	INTEGER	在其上发生此错误的数据库分区号。在非分布式环境中，这是 0。

可以使用 DB2TEXT CLEAR EVENTS 命令来清除事件。

注： 还可在此视图中获取参考事件（例如，启动、落实和完成更新处理）。

在这种情况下，Key1、... Key14 和 OPERATION 都具有空值。

对于视图的索引，PK01、...、PK14 列与在 CREATE INDEX 命令的 KEY COLUMNS 子句中指定的各列相关。

日志表、视图和昵称

日志表的用途就是存储对用户表或视图的更改操作，这些用户表或视图需要与外部全文本索引同步。

对于为常规表或昵称表创建的索引，有一些针对用户表创建的触发器将更改信息填入日志表中。但是，如果使用复制捕获表，那么不创建任何日志表，反而使用复制捕获表。

对于日志表，更新命令读取各个条目，并在成功同步后将它们删除。

但是，对于视图的索引，触发器不能填充日志表。因为可以更新视图，所以用户负责此任务。

表 14. 日志表视图

属性	类型	注释
OPERATION	INTEGER	对需要索引同步的用户表的更改的类型：（0 = 插入、1 = 更新，而 2 = 删除）。
TIME	TIMESTAMP	在此表中创建一行的时间戳记。
PK01 ... PKnm	与用户表相同	出现错误时发生问题的列。对于视图的索引，它们是用户表的主键列或同等键列的副本。

创建该表的用户能够选择、更新、插入和删除此视图。

如果在创建索引命令中指定复制捕获表，那么不创建任何日志表反而使用复制捕获表。复制捕获表必须包含下列各列:

表 15. 复制捕获表

属性	类型	注释
IBMSNAP_OPERATION	INTEGER	对需要索引同步的 CD 或 CCD 表的更改的类型：（I = 插入、U= 更新，而 D= 删除）。 当使用复制捕获表时，更新操作将分割为删除和插入操作。在此情况下，事件表中的插入操作可以来自于对其创建索引的源表的插入或更新操作。
IBMSNAP_COMMITSEQ	CHAR	映射至 CD 或 CCD 表的相应列。
IBMSNAP_INTENTSEQ	CHAR	映射至 CD 或 CCD 表的相应列。
PK01 ... PKnm	与用户表相同	出现错误时发生问题的列。它们是用户表的主键列。

定义该表的用户能够使用授权选项进行选择、更新、插入和删除。

文本搜索引擎原因码

0 操作成功执行 － 没有发生任何错误。

1 将无效句柄传递至函数。

2 函数未能分配足够的内存。

3 由于访问局限性或安全性限制，所以未能执行函数。

4 此版本的文本搜索引擎运行时不支持该操作。

5 当前未启用该操作。

6 应用程序以非法顺序调用文本搜索引擎函数，违反了文本搜索引擎协议。

7 发生了意外错误。请向服务代表报告此错误。

8 指定了无效语言。

9 指定的语言有效，但文本搜索引擎运行时不支持。

10 指定了无效 CCSID。

11 指定的 CCSID 有效，但文本搜索引擎运行时不支持。

12 指定了无效文档标识。

13 指定的文档格式有效，但文本搜索引擎运行时不支持。

14 指定了无效文档格式。

15 由于文件输入/输出期间存在访问局限性，所以该操作失败。

16 由于文件输入/输出期间发生读取错误，所以该操作失败。

17 由于文件输入期间发生读取错误，所以该操作失败。

18 由于文件输出期间发生写错误，所以该操作失败。

19 由于文件输入/输出期间发生寻道错误，所以该操作失败。

20 由于文件输入/输出期间发生通知错误，所以该操作失败。

21 由于文件输入/输出期间发生关闭错误，所以该操作失败。

22 由于重命名操作期间发生错误，所以该操作失败。

23 由于除去操作期间发生错误，所以该操作失败。

24 由于 mkdir 操作期间发生错误，所以该操作失败。

25 一个或多个函数自变量确实具有无效值（例如，意外的空指针或无效枚举类型值）。

26 指定的目录不存在。

27 发生了意外的文本搜索引擎错误。有关进一步的详细信息，请查看错误信息对象中的文本搜索引擎错误代码。

28 发生了意外的 COS 错误。请报告此错误。

29 已尝试更新空文档。

30 此操作不支持指定的自变量。

31 尝试分析日期属性时，日期属性解析器发现无效值。

32 尝试分析编号属性时，编号属性解析器发现无效值。

33 属性名无效，可能是名称过长。

35 保留编号以备将来使用。

36 输入文档包含的属性（DATE、NUMBER 或 STRING）超出属性的长度限制。该属性文本已根据此限制被截断。

38 已超出用户设置的警告阈值。因此生成此错误。

39 未能对输入文档建立索引。它包含过多的嵌套字段。

40 已超出此索引的某个属性类型的不同属性的限制。

46 迭代器不（不再）是有效的，原因是它的列表是空的或已删除。

47 传递的句柄类型不支持该函数。例如，当尝试对并不表示查询结果迭代器的列表迭代器使用 itlQueryResultEntryObtainData 时，会发生此错误。

48 如果针对指定语言和资源路径找不到停用词文件，那么会发出此警告。

49 如果停用词文件不包含任何停用词，那么会发出此警告。

50 如果停用词文件确实包含无效数据，那么会发出此警告。

100 因为具有指定名称和/或目录的索引不存在，所以未能将它打开。

101 指定的索引名是无效的索引名。

102 指定的索引目录是无效的目录名。

103 因为文本搜索引擎检测到索引结构和/或索引文件集已毁坏，所以不能执行该操作。

104 因为具有指定名称和目录的指定索引已存在，所以不能创建。

109 必须执行回滚操作才能对此索引执行任何其他操作。

110 索引配置文件未包含在错误上下文中指定的必需部分。

111 索引配置文件未包含在错误上下文中指定的必需选项。

112 索引配置文件在错误上下文中指定的选项中包含无效数据。

113 索引配置文件与文本搜索引擎版本不匹配。

200 指定的文档模型名是无效的模型名。

201 指定的文档模型字段名是无效的字段名。

202 指定的文档模型未知。

203 指定的文档模型已存在，并且不能重新定义。

204 已向索引添加了过多或过大的文档模型。

205 文档模型包含过多元素。

206 文档模型元素包含此类型元素不允许的参数（XML 属性）。

207 文档模型元素包含此类型参数（XML 属性）不允许的参数值。

208 文档模型元素未包含必需的参数（XML 属性），例如，“name”。

209 文档模型看起来不是 XML，或者它以意外的 XML 元素开头。

210 给定的 XPath（定位器值）包含异常标记。

211 给定的 XPath（定位器值）包含意外的 Axis 说明符（名称后跟两个冒号）。

212 给定的 XPath（定位器值）包含意外的节点测试。

213 文档模型目录文件（extension .mdx）已毁坏。

214 文档模型索引文件（extension .mox）已毁坏。

215 文档包含映射至文档属性的 XML 元素，并且该元素包含另一个文档属性。忽略内部属性。

216 给定的参数值作为 GPP 或 HTML 标记过长。

217 文档模型包含重复的字段定义。

218 文档模型包含重复的属性定义。

300 因为文本搜索引擎检测到用于文档名称映射的索引文件中有毁坏，所以不能执行该操作。

301 因为文本搜索引擎检测到无效文档号，所以不能执行该操作。

302 因为文本搜索引擎检测到无效文档标识，所以不能执行该操作。

303 因为文本搜索引擎找不到文档标识的任何索引条目，所以不能执行该操作。

304 因为文本搜索引擎找不到文档号的任何索引条目，所以不能执行该操作。

305 因为文本搜索引擎检测到使用的文档号中存在溢出，所以不能执行该操作。

306 应用程序尝试对其建立索引的文档标识已出现在文档列表中。文本搜索引擎不支持重复的文档标识出现在一个索引序列中（即，在更新落实之前）。

340 词条强度无效。

341 相关编号无效，必须在有效范围内。

342 相关类型无效，应使用 API 中描述的一个定义。

343 短语（词条）过长。

344 读取时遇到文件的意外结束。

345 读取索引/同义字文件时检测到版本冲突。

346 同义字缓冲区中存在溢出。

347 文件或目录名无效，可能是名称过长。

348 查找没有找到字典中的词条（短语）或定义文件中的条目不包含必需的词条。

349 定义文件是空的。

350 通过输入参数指定的同义字典或定义文件不存在。

351 定义文件中存在语法错误。

352 未正确指定关系。

352 关系编号超出范围。

360 使用了无效的单字符屏蔽。

361 使用了无效的多字符屏蔽。

362 运算符数量小于查询中给定的操作数的数目。

363 运算符值超出 ItlEnOperator 枚举定义的范围。

364 排列公式的值超出枚举范围。

365 标识近似匹配段的编号超出范围。

366 查询正在构造中，不能重定义或复位。

367 作为先前搜索结果给定的作用域表示空结果。

368 在设置第一个字段名之前请求添加字段名的调用无效。

369 忽略请求与索引内容的无效比较的无效搜索标志。例如，如果对以不区分大小写的方式构建的索引请求了区分大小写比较，那么会在错误信息中显示此原因码。

370 泰国语或 DBCS 语言不支持字符串的屏蔽。

371 没有任何有效查询输入。例如，搜索词条可用。

372 请求了无效比较操作。

373 请求了无效比较操作。

374 对空索引请求了搜索索引句柄。

375 运算符和请求的运算符方式的组合不受支持。

380 搜索结果不完整，搜索因为阈值关系而没有继续。

381 索引查询发现查询包含停用词。

401 因为文本搜索引擎检测到用于字段/属性名称映射的索引文件中有毁坏，所以不能执行该操作。

402 因为文本搜索引擎检测到无效字段或属性名，所以不能执行该操作。

403 因为给定的字段或属性名未知，所以不能执行该操作。

404 已超出此索引的某个属性类型的不同属性或不同字段的限制。

500 文档/数据包含无效字符序列（在 UTF8、UTF16 或 DBCS 源代码中）。

501 代码页转换器出错。

502 文档/数据包含不完整的字符序列（在 UTF8、UTF16 或 DBCS 源代码中）。

503 代码页转换器具有无效描述符。

600 XML 文档包含异步实体。例如，未用引号引起来的 XML 属性值。

602 无效字符引用（例如，or）。

603 无效二进制实体引用。

604 未能创建 XML 解析器 Expat。

605 标记中的属性名必须唯一。

607 XML 解析器发现无效外部实体引用。

608 文档包括不正确的标记（例如，缺少 < 或 >）。

609 XML 文档必须具有封闭标记，在此封闭结束标记之后不允许出现任何文本。

610 不允许处理指令出现在其位置上。例如，第一个处理指令不是 prolog <?xml .. ?>.

611 元素由开始标记、内容和结束标记组成。例如，序列“<s> text /s>”中发生此错误，原因是结束标记不正确。

612 XML 解析器中的内存分配失败。

614 无效参数实体引用。

615 不完整字符，可能只是 2 字节 UTF8 字符的第一个字节。

616 递归实体引用。

617 XML 语法错误；例如，文本超出封闭开始标记和结束标记。

618 每一个开始标记都需要相匹配的结束标记。

619 未封闭的数据部分。

620 未封闭的标记；例如，文档中最后一个标记之后的文本。

621 文档中存在未能解析的实体。

622 意外错误。

631 未能对元标记中的字段或属性信息进行语法分析。标记的格式必须是 <meta name=″abc″ content=″xyz″>；可能属性名或元标记的内容不正确。

632 未能将实体变换为字符。

650 不同字段定义以同一开始标记开始。

651 因为一个开始标记包含另一个开始标记，所以这些标记有歧义。

652 如果字段和属性使用同一开始标记，那么它们必须使用同一结束标记或都不使用任何结束标记。

653 文档结束时，字段仍然没有关闭。

654 未对结构化格式指定任何文档模型。会将该文档作为没有字段或属性信息的纯文本文档进行语法分析。

670 因为操作需要“Outside In”（TM）库，而又找不到该库，所以未能执行该操作。

671 因为未能从“Outside In”（TM）库装入必需的过程，所以未能执行该操作。这些库可能过期或已毁坏。

672 使用“Outside In”处理文档时发生错误。

第 10 章 故障诊断

跟踪故障

如果需要向 IBM 代表报告错误，那么可能会要求您打开跟踪以便可将信息写至用于查找错误的文件。

因为跟踪打开时系统性能会受影响，所以仅在 IBM 支持中心代表或您的技术支持代表指导下才应使用跟踪工具。

要打开跟踪，使用 DB2 工具:

```
db2trc on
```

有关进一步的信息，请参阅 DB2 文档。

要接收特定于 Net Search Extender 的信息，可使用带有组件 `96` 的掩码:

```
db2trc on -m *.*.96.*.*
```

在发生严重错误的情况下，查看 `db2diag.log` 可能也有帮助。

删除 DB2 对象（未使用正确 Net Search Extender 命令）

废弃表

在废弃具有一个或多个文本索引的表之前，必须对每个文本索引发出以下命令:

```
db2text drop index <index_name> for text
```

如果在删除索引之前意外地废弃了表，一部分索引将仍然存在，例如，管理表和文本索引文件。

要除去这些文件，使用 `db2text drop index` 命令来删除索引，尽管表已不再存在。

删除数据库

在删除具有一个或多个文本索引的数据库之前，必须对每个文本索引发出以下命令:

```
db2text drop index <index_name> for text
```

如果不使用此命令，那么必须手动删除 `index_directory` 和 `index_work_directory` 的所有索引文件。

注意，如果属于已删除的数据库的索引是在自动更新期间创建的，那么将需要编辑调度程序文件 `ctedem.dat`。

为此，输入以下命令:

对于 UNIX:

```
db2text stop force
cd ~/sqllib/db2ext
```

对于 Windows:

```
db2text stop force
cd <db2_install_path>\sqllib\<db2_instance_name>\db2ext
```

打开目录中的 ctedem.dat 文件，并除去涉及到已删除的数据库的所有条目。

Windows 上的安装返回码

Windows 上的 setup.exe 返回码

setup.log 中的 setup.exe 返回码如下所示:

- 0 成功
- -1 一般错误
- -2 方式无效
- -3 在 setup.iss 文件中未找到所需的数据
- -4 没有足够的可用内存
- -5 文件不存在
- -6 不能写入响应文件
- -7 无法写入日志文件
- -8 Install Shield Silent（.iss）响应文件的路径无效
- -9 不是有效列表类型
- -10 数据类型无效
- -11 安装期间发生未知错误
- -12 对话框的顺序错误
- -51 无法创建指定的文件夹
- -52 无法访问指定的文件或文件夹
- -53 选择的选项无效

提示和小技巧

权限 在 Windows 上发出 DB2TEXT START 命令时，应确保您是 Administrators 组的成员。否则，DB2TEXT START 命令将失败并且返回以下消息：CTE0218 函数 "OpenSCManager()"失败，错误代码为"5"。

权限 在 Windows 上。确保使用使用用户帐户而不是系统帐户来运行 Net Search Extender 实例服务 DB2EXT-<DB2_instance_name>。如果使用系统帐户来运行该实例服务，那么将无法启用数据库。

语言 由于 Net Search Extender 事件日志消息始终是以 DB2 服务器语言显示的，因此，对于从 DB2 控制中心发出的命令所产生的事件日志消息，可能会采用与为 DB2 控制中心设置的语言不相同的语言来显示。

修改 db2cli.ini 文件

如果您修改了 db2cli.ini 文件，但是在使用 Net Search Extender 时发生了问题，那么应复原原始版本的 db2cli.ini。

客户机与服务器之间的互操作性

必须在服务器和客户机上安装相同级别的 Net Search Extender 修订包。

只有受支持的平台才能实现客户机与服务器之间的互操作性。

日志大小

如果由于错误消息和警告消息需要的空间超过了可用的 DB2 日志大小而导致未完成索引建立过程，那么 DB2 将回滚整个事务而不落实逻辑日志表条目。这就意味着您将看不到这些条目。

有关如何增大事务日志大小以避免发生上述情况的信息，请参阅 DB2 文档。

DBCS 对象名

如果在 db2text 管理命令中使用了 DBCS 对象名，那么需要将这些名称用双引号引起来，以避免将它们转换为大写。

对昵称进行增量索引更新

如果同时对两个或多个索引启动初始更新，那么 Update 命令可能会返回 SQL0803N 错误。在此情况下，请尝试再次执行 Update 命令。

单个字符屏蔽和字符规范化

诸如 'über' 这样的词语是规范的，按它们的规范格式（'ueber'）存储在索引中。因此，如果发出一个包含单个字符屏蔽的查询（例如，'_ber'），那么会找不到 'über'。

使用重复的高速缓存列名

如果使用重复的高速缓存列名，在创建文本索引或更新索引期间不会出错，但是将无法进行搜索。当您尝试进行搜索时，将产生 SQL 错误消息，指出使用了重复列。

不正确的共享内存大小

如果在 db2text activate cache 命令中指定的最大高速缓存大小太小，那么在产生的错误消息中所显示的必需高速缓存大小是不正确的。

请使用 DB2EXT.MAXIMUM_CACHE_SIZE 和 DB2EXT.PCTFREE 函数来检查正确的高速缓存大小。请使用 db2text alter index 命令来更正最大高速缓存大小，然后再次激活高速缓存。

非 Unicode 数据库中的 Unicode 表

如果数据库不支持 Unicode，将不允许对 Unicode 表创建文本索引。

在 Linux 上，在查询中对 LANG 变量使用了不正确的代码页

如果在查询时对 LANG 变量设置使用 7 位 ASCII 代码页，那么会显示以下错误：SearchString parse: check LANG & locale charmap values.。

为了避免发生这种情况，应将 LANG 变量更改为 8 位 LANG 值，重新启动 DB2，然后再次尝试进行搜索。

文件访问问题

如果无法访问文件（例如，文档模型、文本索引或辞典），那么请确保您使用了正确的密码并且具有正确的权限来运行 Net Search Extender 实例服务。这特别适用于已映射的网络驱动器上的共享资源。

无法使用高速缓存

在搜索或激活高速缓存期间，可能会显示以下错误消息：CTE0271 不可使用高速缓存，需要执行 DEACTIVATE 和 ACTIVATE RECREATE。要解决此问题，请检查系统设置并尝试增加调页空间和可用内存量。

在卸载之后并未删除实例服务

如果在卸载 Net Search Extender 之后并未删除实例服务，那么使用以下工具来手动删除这些服务：`ctereg <instancename> unregister`。例如，`ctereg db2-0 unregister`。

在 UNIX 上，cteprcrx 异常终止

检查所使用的实例所有者是否没有单独的受防护用户标识。要进行此检查，打开 `<instance_home_dir>/sqllib/adm` 中的 `.fenced` 文件，并检查实例所有者是否也是受防护用户。

附录 A. DB2 技术信息概述

可以通过下列工具和方法获取 DB2 技术信息:

- *DB2 信息中心*
  - 主题（任务、概念和参考主题）
  - DB2 工具的帮助
  - 样本程序
  - 教程
- DB2 书籍
  - PDF 文件（可下载）
  - PDF 文件（在 DB2 PDF DVD 中）
  - 印刷版书籍
- 命令行帮助
  - 命令帮助
  - 消息帮助

注: *DB2 信息中心*主题的更新频率比 PDF 书籍或硬拷贝书籍的更新频率高。要获取最新信息，请安装可用的文档更新，或者参阅 ibm.com® 上的 *DB2 信息中心*。

可以在线访问 ibm.com 上的其他 DB2 技术信息，如技术说明、白皮书和 IBM Redbooks® 出版物。访问位于以下网址的 DB2 信息管理软件库站点: http://www.ibm.com/software/data/sw-library/。

文档反馈

我们非常重视您对 DB2 文档的反馈。如果您想就如何改善 DB2 文档提出建议，请发送电子邮件至 db2docs@ca.ibm.com。DB2 文档小组会阅读您的所有反馈，但不能直接答复您。请尽可能提供具体的示例，这样我们才能更好地了解您所关心的问题。如果您要提供有关具体主题或帮助文件的反馈，请加上标题和 URL。

请不要用以上电子邮件地址与 DB2 客户支持机构联系。如果您遇到文档不能解决的 DB2 技术问题，请与您当地的 IBM 服务中心联系以获得帮助。

如果愿意帮助 IBM 使 IBM 信息管理产品更易于使用，请参与可消费性调查: http://www.ibm.com/software/data/info/consumability-survey/。

硬拷贝或 PDF 格式的 DB2 技术库

下列各表描述 IBM 出版物中心（网址为 www.ibm.com/shop/publications/order）提供的 DB2 资料库。可以从 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 下载 PDF 格式的英文 DB2 版本 9.5 手册和已翻译的版本。

尽管这些表标识书籍有印刷版，但可能未在您所在国家或地区提供。

每次更新手册时，表单号都会递增。确保您正在阅读下面列示的手册的最新版本。

注: *DB2* 信息中心的更新频率比 PDF 或硬拷贝书籍的更新频率高。

表 16. DB2 技术信息

书名	书号	是否提供印刷版	最近一次更新时间
Administrative API Reference	SC23-5842-02	是	2009 年 4 月
Administrative Routines and Views	SC23-5843-02	否	2009 年 4 月
Call Level Interface Guide and Reference, Volume 1	SC23-5844-02	是	2009 年 4 月
Call Level Interface Guide and Reference, Volume 2	SC23-5845-02	是	2009 年 4 月
Command Reference	SC23-5846-02	是	2009 年 4 月
《数据移动指南和参考》	S151-0617-01	是	2009 年 4 月
《数据恢复及高可用性指南与参考》	S151-0619-01	是	2009 年 4 月
《数据服务器、数据库和数据库对象指南》	S151-0612-01	是	2009 年 4 月
《数据库安全性指南》	S151-0614-01	是	2009 年 4 月
Developing ADO.NET and OLE DB Applications	SC23-5851-02	是	2009 年 4 月
Developing Embedded SQL Applications	SC23-5852-02	是	2009 年 4 月
Developing Java™ Applications	SC23-5853-02	是	2009 年 4 月
Developing Perl and PHP Applications	SC23-5854-02	否	2009 年 4 月
Developing User-defined Routines (SQL and External)	SC23-5855-02	是	2009 年 4 月
Getting Started with Database Application Development	GC23-5856-02	是	2009 年 4 月
《Linux 和 Windows 上的 DB2 安装和管理入门》	G151-0623-01	是	2009 年 4 月
《国际化指南》	S151-0616-01	是	2009 年 4 月
《消息参考，第 1 卷》	G151-0632-01	否	2009 年 4 月
《消息参考，第 2 卷》	G151-0633-01	否	2009 年 4 月
《迁移指南》	G151-0622-01	是	2009 年 4 月
《Net Search Extender 管理和用户指南》	S151-0760-01	是	2009 年 4 月
《分区和集群指南》	S151-0615-01	是	2009 年 4 月
Query Patroller Administration and User's Guide	SC23-8507-01	是	2009 年 4 月

表 16. DB2 技术信息 (续)

书名	书号	是否提供印刷版	最近一次更新时间
《IBM 数据服务器客户机快速入门》	G151-0625-01	否	2009 年 4 月
《DB2 服务器快速入门》	G151-0624-01	是	2009 年 4 月
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC23-8508-02	是	2009 年 4 月
SQL Reference, Volume 1	SC23-5861-02	是	2009 年 4 月
SQL Reference, Volume 2	SC23-5862-02	是	2009 年 4 月
《系统监视器指南和参考》	S151-0618-01	是	2009 年 4 月
Text Search Guide	SC23-5866-01	是	2009 年 4 月
《故障诊断指南》	G151-0621-01	否	2009 年 4 月
《调整数据库性能》	S151-0613-01	是	2009 年 4 月
《Visual Explain 教程》	S151-0634-00	否	
《新增内容》	S151-0629-01	是	2009 年 4 月
Workload Manager Guide and Reference	SC23-5870-02	是	2009 年 4 月
《pureXML 指南》	S151-0630-01	是	2009 年 4 月
《XQuery 参考》	S151-0631-01	否	2009 年 4 月

表 17. 特定于 DB2 Connect 的技术信息

书名	书号	是否提供印刷版	最近一次更新时间
《DB2® Connect™ 个人版快速入门》	G151-0627-01	是	2009 年 4 月
《DB2 Connect 服务器快速入门》	G151-0628-01	是	2009 年 4 月
《DB2 Connect 用户指南》	S151-0626-01	是	2009 年 4 月

表 18. Information Integration 技术信息

书名	书号	是否提供印刷版	最近一次更新时间
Information Integration: Administration Guide for Federated Systems	SC19-1020-01	是	2008 年 3 月
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-02	是	2008 年 3 月
Information Integration: 《联合数据源配置指南》	S151-0468-00	否	

表 18. Information Integration 技术信息 (续)

书名	书号	是否提供印刷版	最近一次更新时间
Information Integration:《SQL 复制指南和参考》	S151-0475-00	是	2008 年 3 月
Information Integration: Introduction to Replication and Event Publishing	SC19-1028-01	是	2008 年 3 月

订购印刷版的 DB2 书籍

如果您需要印刷版的 DB2 书籍，可以在许多（但不是所有）国家或地区在线购买。无论何时都可以从当地的 IBM 代表处订购印刷版的 DB2 书籍。请注意，*DB2 PDF 文档* DVD 上的某些软拷贝书籍没有印刷版。例如，*DB2 消息参考*的任何一卷都没有提供印刷版书籍。

只要支付一定费用，就可以从 IBM 获取 DB2 PDF 文档 DVD，该 DVD 包含许多印刷版 DB2 书籍的内容。根据您下订单的位置，您可能能够从 IBM 出版物中心在线订购书籍。如果在线订购在您所在国家或地区不可用，您始终可以从当地的 IBM 代表处订购印刷版 DB2 书籍。注意，并非 DB2 PDF 文档 DVD 上的所有书籍都有印刷版。

注：最新最完整的 DB2 文档保留在 DB2 信息中心中，网址如下：http://publib.boulder.ibm.com/infocenter/db2luw/v9r5。

要订购印刷版的 DB2 书籍:

- 要了解您是否可从所在国家或地区在线订购印刷版的 DB2 书籍，可查看 IBM 出版物中心站点，网址为：http://www.ibm.com/shop/publications/order。必须先选择国家、地区或语言才能访问出版物订购信息，然后再按照针对您所在位置的订购指示信息进行订购。
- 要从当地的 IBM 代表处订购印刷版的 DB2 书籍:
  1. 从下列其中一个 Web 站点找到当地代表处的联系信息:
     - IBM 全球联系人目录，网址为 www.ibm.com/planetwide。
     - IBM 出版物 Web 站点，网址为 http://www.ibm.com/shop/publications/order。必须先选择国家、地区或语言才能访问对应您的所在地的出版物主页。在此页面中访问“关于此站点”链接。
  2. 请在致电时说明您想订购 DB2 出版物。
  3. 请向您当地的代表提供想要订购的书籍的书名和书号。有关书名和书号的信息，请参阅第 217 页的『硬拷贝或 PDF 格式的 DB2 技术库』。

从命令行处理器显示 SQL 状态帮助

DB2 返回描述 SQL 语句执行结果的 SQLSTATE。SQLSTATE 帮助说明 SQL 状态和 SQL 状态类代码的含义。

要调用 SQL 状态帮助，请打开命令行处理器并输入:

```
? sqlstate or ? class code
```

其中，*sqlstate* 表示有效的 5 位 SQL 状态，*class code* 表示该 SQL 状态的前 2 位。例如，`? 08003` 显示 08003 SQL 状态的帮助，而 `? 08` 显示 08 类代码的帮助。

访问不同版本的 DB2 信息中心

对于 DB2 版本 9.5 主题，DB2 信息中心 URL 为 http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/。

对于 DB2 版本 9 主题，DB2 信息中心 URL 为 http://publib.boulder.ibm.com/infocenter/db2luw/v9/

对于 DB2 版本 8 主题，请访问以下版本 8 信息中心 URL: http://publib.boulder.ibm.com/infocenter/db2luw/v8/

在 DB2 信息中心中以您的首选语言显示主题:

DB2 信息中心尝试以您在浏览器首选项中指定的语言显示主题。如果未提供主题的首选语言翻译版本，那么 DB2 信息中心将显示该主题的英文版。

- 要在 Internet Explorer 浏览器中以您的首选语言显示主题:
  1. 在 Internet Explorer 中，单击**工具** —> **Internet 选项** —> **语言...**按钮。"语言首选项"窗口打开。
  2. 确保您的首选语言被指定为语言列表中的第一个条目。
     - 要将新语言添加至列表，请单击**添加...** 按钮。

       注: 添加语言并不能保证计算机具有以首选语言显示主题所需的字体。
     - 要将语言移至列表顶部，请选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一个条目。
  3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。
- 要在 Firefox 或 Mozilla 浏览器中以首选语言显示主题:
  1. 在**工具** —> **选项** —> **高级**对话框中的**语言**部分中选择按钮。"语言"面板将显示在"首选项"窗口中。
  2. 确保您的首选语言被指定为语言列表中的第一个条目。
     - 要将新语言添加至列表，请单击**添加...** 按钮以从"添加语言"窗口中选择一种语言。
     - 要将语言移至列表顶部，请选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一个条目。
  3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。

在某些浏览器和操作系统组合上，可能还必须将操作系统的区域设置更改为您选择的语言环境和语言。

更新安装在您的计算机或内部网服务器上的 DB2 信息中心

如果已经在本地安装了 DB2 信息中心，那么您可以从 IBM 获取文档更新并安装。

要更新在本地安装的 *DB2 信息中心*，您需要：

1. 停止计算机上的 *DB2 信息中心*，然后以独立方式重新启动信息中心。如果以独立方式运行信息中心，那么网络上的其他用户将无法访问信息中心，因而您可以应用更新。非管理员和非 Root 用户 *DB2 信息中心*始终以独立方式运行。。
2. 使用"更新"功能部件来查看可用的更新。如果有您希望安装的更新，那么请使用"更新"功能部件来获取并安装这些更新。

   注： 如果您所在的环境要求在一台未连接至因特网的机器上安装 *DB2 信息中心*更新，那么必须使用一台已连接至因特网的机器将更新站点镜像至本地文件系统，并安装 *DB2 信息中心*。如果网络中有许多用户将安装文档更新，那么可以通过在本地也为更新站点制作镜像并为更新站点创建代理来缩短每个人执行更新所需要的时间。

   如果提供了更新包，请使用"更新"功能部件来获取这些更新包。但是，只有在独立方式下才能使用更新功能。
3. 停止独立信息中心，然后在计算机上重新启动 *DB2 信息中心*。

注： 在 Windows Vista 上，必须以管理员身份才能运行下面所列示的命令。要启动具有全面管理员特权的命令提示符或图形工具，请右键单击快捷方式，然后选择**以管理员身份运行**。

要更新安装在您的计算机或内部网服务器上的 *DB2 信息中心*：

1. 停止 *DB2 信息中心*。
   - 在 Windows 上，单击**开始** → **控制面板** → **管理工具** → **服务**。右键单击 **DB2 信息中心**服务，并选择**停止**。
   - 在 Linux 上，输入以下命令：

     ```
     /etc/init.d/db2icdv95 stop
     ```
2. 以独立方式启动信息中心。
   - 在 Windows 上：
     a. 打开命令窗口。
     b. 浏览至信息中心的安装位置。缺省情况下，*DB2 信息中心*安装在 *Program_files*\IBM\DB2 Information Center\Version 9.5 目录中，其中 *Program_files* 表示 Program Files 目录的位置。
     c. 从安装目录浏览至 doc\bin 目录。
     d. 运行 help_start.bat 文件：

        ```
        help_start.bat
        ```
   - 在 Linux 上：
     a. 浏览至信息中心的安装位置。缺省情况下，*DB2 信息中心*安装在 /opt/ibm/db2ic/V9.5 目录中。
     b. 从安装目录浏览至 doc/bin 目录。

c. 运行 help_start 脚本:

```
help_start
```

系统缺省 Web 浏览器将启动以显示独立信息中心。

3. 单击**更新**按钮（ ）。在信息中心的右边面板上，单击**查找更新**。 将显示现有文档的更新列表。
4. 要启动安装进程，请检查您要安装的选项，然后单击**安装更新**。
5. 在安装进程完成后，请单击**完成**。
6. 要停止独立信息中心，请执行下列操作:
   - 在 Windows 上，浏览至安装目录的 doc\bin 目录并运行 help_end.bat 文件:

     ```
     help_end.bat
     ```

     注: help_end 批处理文件包含安全地终止使用 help_start 批处理文件启动的进程所需的命令。不要使用 `Ctrl-C` 或任何其他方法来终止 help_start.bat。
   - 在 Linux 上，浏览至安装目录的 doc/bin 目录并运行 help_end 脚本:

     ```
     help_end
     ```

     注: help_end 脚本包含安全地终止使用 help_start 脚本启动的进程所需的命令。不要使用任何其他方法来终止 help_start 脚本。
7. 重新启动 *DB2 信息中心*。
   - 在 Windows 上，单击**开始** → **控制面板** → **管理工具** → **服务**。右键单击 **DB2 信息中心**服务，并选择**启动**。
   - 在 Linux 上，输入以下命令:

     ```
     /etc/init.d/db2icdv95 start
     ```

更新后的 *DB2 信息中心*将显示新的以及更新后的主题。

DB2 教程

DB2 教程帮助您了解 DB2 产品的各个方面。这些课程提供了逐步指示信息。

开始之前

可从信息中心查看 XHTML 版的教程: http://publib.boulder.ibm.com/infocenter/db2help/ 。

某些课程使用了样本数据或代码。有关其特定任务的任何先决条件的描述，请参阅教程。

DB2 教程

要查看教程，请单击标题。

《*pureXML 指南*》中的『**pureXML™**』
: 设置 DB2 数据库以存储 XML 数据以及对本机 XML 数据存储执行基本操作。

《*Visual Explain 教程*》中的『**Visual Explain**』
: 使用 Visual Explain 来分析、优化和调整 SQL 语句以获取更好的性能。

DB2 故障诊断信息

提供了很多故障诊断和问题确定信息以帮助您使用 DB2 数据库产品。

DB2 文档

故障诊断信息可在 DB2 信息中心的“DB2 故障诊断指南”或“数据库基础”部分中找到。可在该处找到有关如何使用 DB2 诊断工具和实用程序来隔离和找出问题的信息、某些最常见问题的解决方案以及有关如何解决使用 DB2 数据库产品时可能遇到的问题的建议。

DB2 技术支持 Web 站点

如果您遇到了问题并且想要获取查找可能的原因和解决方案的帮助，请参阅 DB2 技术支持 Web 站点。该“技术支持”站点具有指向最新 DB2 出版物、技术说明、授权程序分析报告（APAR 或错误修订）、修订包和其他资源的链接。可搜索此知识库并查找问题的可能解决方案。

请访问 DB2 技术支持 Web 站点：http://www.ibm.com/software/data/db2/support/db2_9/。

条款和条件

如果符合以下条款和条件，那么授予您使用这些出版物的准用权。

个人使用：只要保留所有的专有权声明，您就可以为个人、非商业使用复制这些出版物。未经 IBM 明确同意，您不可以分发、展示或制作这些出版物或其中任何部分的演绎作品。

商业使用：只要保留所有的专有权声明，您就可以仅在企业内复制、分发和展示这些出版物。未经 IBM 明确同意，您不可以制作这些出版物的演绎作品，或者在您的企业外部复制、分发或展示这些出版物或其中的任何部分。

除非本准用权中有明确授权，不得把其他准用权、许可或权利（无论是明示的还是暗含的）授予其中包含的出版物或任何信息、数据、软件或其他知识产权。

当使用这些出版物损害了 IBM 的利益，或者根据 IBM 的规定，未正确遵守上述指导说明时，那么 IBM 保留自主决定撤销本文授予的准用权的权利。

您不可以下载、出口或再出口本信息，除非完全遵守所有适用的法律和法规，包括所有美国出口法律和法规。

IBM 对这些出版物的内容不作任何保证。这些出版物“按现状”提供，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的关于适销和适用于某种特定用途的保证。

附录 B. 声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，那么由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区: International Business Machines Corporation"按现状"提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本文档可能提供非 IBM Web 站点和资源的链接或引用。IBM 对于任何非 IBM Web 站点或第三方资源不作任何声明、保证或其他承诺，即使本文档可能引用了这些 Web 站点或第三方资源，或者可从本文档访问或链接到这些 Web 站点或第三方资源。到某个非 IBM Web 站点的链接并不意味着 IBM 认可此类 Web 站点的内容或使用或其所有者。此外，IBM 不是您与任何第三方签署协议的任何交易的一方，也不对任何交易负责，即使您从某个 IBM 站点了解到此类第三方或使用到此类第三方的链接时亦如此。因此，您需要承认并同意，IBM 不对此类外部站点或资源的可用性负责，也不对可从那些站点或资源上获得的任何内容、服务、产品或其他资料承担任何责任或义务。第三方提供的任何软件须遵守该软件随附的许可证的条款和条件。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息可能包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可:

本信息可能包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明:

©（*贵公司的名称*）（*年份*）. 此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp. （*输入年份*）. All rights reserved.

商标

IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp. 在全世界许多管辖区域的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。在版权和商标信息的 Web 页面上提供了 IBM 商标的当前列表，网址为：www.ibm.com/legal/copytrade.shtml。

下列术语是其他公司的商标或注册商标。

- Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。
- Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和/或其他国家或地区的商标。
- UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。
- Intel、Intel 徽标、Intel Inside、Intel Inside 徽标、Intel Centrino、Intel Centrino 徽标、Celeron、Intel Xeon、Intel SpeedStep、Itanium 和 Pentium 是 Intel Corporation 或其子公司在美国和其他国家或地区的商标或注册商标。Intel 商标信息
- Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[W]

[X]

[Y]

[Z]

A

O

S

U

V

W

X

IBM

中国印刷

S151-0760-01



Spine information:

DB2 版本 **9.5 Linux** 版、**UNIX** 版和 **Windows** 版

Net Search Extender 管理和用户指南

IBM